

# Type Soundness for Path Polymorphism<sup>★</sup>

Andrés Viso<sup>a,1</sup> Eduardo Bonelli<sup>b,2</sup> Mauricio Ayala-Rincón<sup>c,3</sup>

<sup>a</sup> *Consejo Nacional de Investigaciones Científicas y Técnicas – CONICET  
Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires – UBA  
Buenos Aires, Argentina*

<sup>b</sup> *Consejo Nacional de Investigaciones Científicas y Técnicas – CONICET  
Departamento de Ciencia y Tecnología  
Universidad Nacional de Quilmes – UNQ  
Bernal, Argentina*

<sup>c</sup> *Departamentos de Matemática e Ciência da Computação  
Universidade de Brasília – UnB  
Brasília D.F., Brasil*

---

## Abstract

*Path polymorphism* is the ability to define functions that can operate uniformly over arbitrary recursively specified data structures. Its essence is captured by patterns of the form  $x\ y$  which decompose a compound data structure into its parts. Typing these kinds of patterns is challenging since the type of a compound should determine the type of its components. We propose a static type system (*i.e.* no run-time analysis) for a pattern calculus that captures this feature. Our solution combines type application, constants as types, union types and recursive types. We address the fundamental properties of Subject Reduction and Progress that guarantee a well-behaved dynamics. Both these results rely crucially on a notion of *pattern compatibility* and also on a coinductive characterisation of subtyping.

*Keywords:*  $\lambda$ -Calculus, Pattern Matching, Path Polymorphism, Static Typing

---

## 1 Introduction

Applicative representation of data structures in functional programming languages consists in applying variable arity constructors to arguments. Examples are:

$$\begin{aligned} s &= \text{cons}(\text{vl } v_1) (\text{cons}(\text{vl } v_2) \text{nil}) \\ t &= \text{node}(\text{vl } v_3) (\text{node}(\text{vl } v_4) \text{nil nil}) (\text{node}(\text{vl } v_5) \text{nil nil}) \end{aligned}$$

These are data structures that hold values, prefixed by the constructor `vl` for “value” ( $v_{1,2}$  in the first case, and  $v_{3,4,5}$  in the second). Consider the following function for

---

<sup>★</sup> Work partially funded by the international projects DeCOPA STIC-AmSud 146/2012, CONICET, CAPES, CNRS; and ECOS-Sud A12E04, CONICET, CNRS.

<sup>1</sup> Email: [aeviso@dc.uba.ar](mailto:aeviso@dc.uba.ar)

<sup>2</sup> Email: [eabonelli@gmail.com](mailto:eabonelli@gmail.com)

<sup>3</sup> Email: [ayala@unb.br](mailto:ayala@unb.br)

updating the values of any of these two structures by applying some user-supplied function  $f$  to it:

$$\text{upd} = f \rightarrow_{\{f:A \supset B\}} \left( \begin{array}{l} \text{vl } z \rightarrow_{\{z:A\}} \quad \text{vl } (f z) \\ | \quad x y \rightarrow_{\{x:C, y:D\}} (\text{upd } f x) (\text{upd } f y) \\ | \quad w \rightarrow_{\{w:E\}} \quad w \end{array} \right) \quad (1)$$

Both  $\text{upd } (+1) s$  and  $\text{upd } (+1) t$  may be evaluated. The expression to the right of “=” is called an *abstraction* and consists of a unique *branch*; this branch in turn is formed from a pattern ( $f$ ), a user-specified type declaration for the variables in the pattern ( $\{f : A \supset B\}$ ), and a body (in this case the body is itself another abstraction that consists of three branches). An argument to an abstraction is matched against the patterns, in the order in which they are written, and the appropriate body is selected. Notice the pattern  $xy$ . This pattern embodies the essence of what is known as *path polymorphism* [17, 19] since it abstracts a path being “split”. The starting point of this paper is how to type a calculus, let us call it **CAP** for *Calculus of Applicative Patterns*, that admits such examples. We next show why the problem is challenging, explain our contribution and also discuss why the current literature falls short of addressing it. We do so with an introduction-by-example approach, for the full syntax and semantics of the calculus refer to Sec. 2.

### Preliminaries on typing patterns expressing path polymorphism

Consider these two simple examples:

$$(\text{nil} \rightarrow 0) \text{ cons} \qquad (\text{vl } x \rightarrow_{\{x:\text{Nat}\}} x + 1) (\text{vl true}) \quad (2)$$

They should clearly not be typable. In the first case, the abstraction is not capable of handling **cons**. This is avoided by introducing singleton types in the form of the constructors themselves: **nil** is given type  $\text{nil}$  while **cons** is given type **cons**; these are then compared. In the second case,  $x$  in the pattern is required to be **Nat** yet the type of the argument to **vl** in  $\text{vl true}$  is **Bool**. This is avoided by introducing type application [24] into types:  $\text{vl } x$  is assigned a type of the form  $\text{vl } @ \text{ Nat}$  while  $\text{vl true}$  is assigned type  $\text{vl } @ \text{ Bool}$ ; these are then compared.

Consider next the pattern  $xy$  of **upd**. It can be instantiated with different applicative terms in each recursive call to **upd**. For example, suppose  $A = B = \text{Nat}$ , that  $v_1$  and  $v_2$  are numbers and consider  $\text{upd } (+1) s$ . The following table illustrates some of the terms with which  $x$  and  $y$  are instantiated during the evaluation of  $\text{upd } (+1) s$ :

|   | $x$                             | $y$  |
|---|---------------------------------|--|
| $\text{upd } (+1) s$  | <b>cons</b> ( <b>vl</b> $v_1$ ) | <b>cons</b> ( <b>vl</b> $v_2$ ) <b>nil</b> |
| $\text{upd } (+1) (\text{cons } (\text{vl } v_1))$            | <b>cons</b>                     | <b>vl</b> $v_1$                            |
| $\text{upd } (+1) (\text{cons } (\text{vl } v_2) \text{nil})$ | <b>cons</b> ( <b>vl</b> $v_2$ ) | <b>nil</b>                                 |

The type assigned to  $x$  (and  $y$ ) should encompass all terms in its respective column. This suggests adopting a union type for  $x$ . On the assumption that the programmer

has provided an exhaustive coverage, the type of  $x$  in **upd** is:

$$\mu\alpha.(\text{vl} @ A) \oplus (\alpha @ \alpha) \oplus (\text{cons} \oplus \text{node} \oplus \text{nil})$$

Here  $\mu$  is the recursive type constructor and  $\oplus$  the union type constructor. The variable  $y$  in the pattern  $xy$  will also be assigned the same type. Note that **upd** itself is assigned type  $(A \supset B) \supset (F_A \supset F_B)$ , where  $F_X$  is  $\mu\alpha.(\text{vl} @ X) \oplus (\alpha @ \alpha) \oplus (\text{cons} \oplus \text{node} \oplus \text{nil})$ . Thus variables in applicative patterns will be assigned union types.

Recursive types are useful to give static semantics to fixpoint combinators, which embodies the essence of recursion and thus *path polymorphism*. Together with unions, they allow to model recursively defined data types. Combining these ideas with type application allows to define data types in a more intuitive manner, like for example lists and trees

$$\mu\alpha.\text{nil} \oplus (\text{cons} @ A @ \alpha) \quad \mu\alpha.\text{nil} \oplus (\text{node} @ A @ \alpha @ \alpha)$$

The advantage of this approach is that the type expression reflects the structure of the terms that inhabit it (*cf.* Fig. 5). This will prove to be convenient for our proposed notion of *pattern compatibility*.

Compatibility is the key for ensuring Safety (Subject Reduction, SR for short, and Progress). Consider the following example:

$$(\text{vl } x \rightarrow_{\{x:\text{Bool}\}} \text{if } x \text{ then } 1 \text{ else } 0) \mid (\text{vl } y \rightarrow_{\{y:\text{Nat}\}} y + 1) \quad (3)$$

Although there is a branch capable of handling a term such as **vl 4**, namely the second one, evaluation in **CAP** takes place in left-to-right order following standard practice in functional programming languages. Since the term **vl 4** *also* matches the pattern **vl x**, we would obtain the (incorrect) reduct **if 4 then 1 else 0**. We thus must relate the types of **vl x** and **vl y** in order to avoid failure of SR. Since **vl y** is an instance of **vl x**, we require the type of the latter to be a subtype of the type of the former since it will always have priority:  $\text{vl} @ \text{Nat} \preceq \text{vl} @ \text{Bool}$ . Fortunately, this is not the case since  $\text{Nat} \not\preceq \text{Bool}$ , rendering this example untypable.

Consider now, a term such as:

$$\begin{aligned} f \rightarrow_{\{f:A \supset B\}} & \left( \text{vl } z \rightarrow_{\{z:A\}} \quad \text{vl } (f z) \right. \\ & \left. \mid x y \rightarrow_{\{x:C, y:D\}} x y \right) \end{aligned} \quad (4)$$

This function takes an argument  $f$  and pattern-matches with a data structure to apply  $f$  only when this data structure is an application with the constructor **vl** on the left-hand side. Assigning  $x$  in the second branch the type  $C = \text{vl}$  is a potential source of failure of SR since the function would accept arguments of type  $\text{vl} @ D$ . Our proposed notion of compatibility will check the *types* occurring at offending positions in the *types* of both patterns. In this case, if  $C = \text{vl}$  then  $C @ D \preceq \text{vl} @ A$  is enforced. Note that if  $C$  were a type such as  $\mu\alpha.\text{vl} \oplus \alpha @ \alpha$ , then also the same condition would be enforced.

Let us return to example (1). The type declarations would be  $C = D = \mu\alpha.(\text{vl} @ A) \oplus (\alpha @ \alpha) \oplus (\text{cons} \oplus \text{node} \oplus \text{nil})$  and  $E = \text{cons} \oplus \text{node} \oplus \text{nil}$ . We now

illustrate how compatibility determines any possible source of failure of SR. Let us call  $p, q$  and  $r$  the three patterns of the innermost abstraction of (1), resp. Since pattern  $p$  does not subsume  $q$ , we determine the (maximal) positions in both patterns which are sources of failure of subsumption. In this case, it is that of  $\forall l$  in  $p$  and  $x$  in  $q$ . We now consider the *subtype* at that position in  $\forall l @ A$ , the type of  $p$ , and the *subtype* at the same position in  $F_A @ F_A$ , the type of  $q$ : the first is  $\forall l$  and the second is  $F_A$ . Since  $F_A$  does not admit  $\forall l$  (cf. Def. 3.37), these branches are immediately declared compatible. In the case of  $p$  and  $r$ ,  $\epsilon$  is the offending position in the failure of  $p$  subsuming  $r$ : since the type application constructor  $@$  located at position  $\epsilon$  in  $\forall l @ A$  is not admitted by  $E$ , the type of  $r$ , these branches are immediately declared compatible. Finally, a similar analysis between  $q$  and  $r$  entails that these are compatible too. The type system and its proof of Safety will therefore assure us that this example preserves typability.

### Summary of contributions:

- A typing discipline for CAP. We statically guarantee safety for path polymorphism in its purest form (other, more standard forms of polymorphism such as parametric polymorphism which we believe to be easier to handle, are out of the scope of this paper).
- A proof of safety for the resulting system. It relies on the syntactic notion of pattern compatibility mentioned above, hence no runtime analysis is required.
- Invertibility of subtyping of recursive types. This is crucial for the proof of safety. It relies on an equivalent coinductive formulation for which invertibility implies invertibility of subtyping of recursive types.

### Related work

The literature on (typed) pattern calculi is extensive; we mention the most relevant ones (see [17, 19] for a more thorough listing). In [2] the constructor calculus is proposed. It has a different notion of pattern matching: it uses a case construct  $\{c_1 \mapsto s_1, \dots, c_n \mapsto s_n\} \cdot t$  in which certain occurrences of the constructors  $c_i$  in  $t$  are replaced by their corresponding terms. [24] studies typing to ensure that these constructor substitutions never block on a constant not in their domain. Recursive types are not considered (nor is path polymorphism). Two further closely related efforts merit comments: the first is the work by Jay and Kesner and the second is that of the  $\rho$ -calculus by Kirchner and colleagues.

In [18, 19] the Pure Pattern Calculus (PPC) is studied. It allows patterns to be computed dynamically (they may contain free variables). A type system for a PPC like calculus is given in [17] however neither recursive nor union types are considered. [17] also studies a simple static pattern calculus. However, there are numerous differing aspects w.r.t. this work among which we can mention the following. First, the typed version of [17] (the *Query Calculus*) omits recursive types and union types. Then, although it admits a form of path polymorphism, this is at the cost of matching types at runtime and thus changing the operational semantics of the untyped calculus; our system is purely static, no runtime analysis is required.

The  $\rho$ -calculus [10] is a generic pattern matching calculus parameterized over a

matching theory. There has been extensive work exploring numerous extensions [5, 11–14, 22]. None addresses path polymorphism however. Indeed, none of the above allow patterns of the form  $xy$ . This limitation seems to be due to the alternative approach to typing  $\mathbf{c}x$  adopted in the literature on the  $\rho$ -calculus where  $\mathbf{c}$  is assigned a *fixed* functional type. This approach seems incompatible with path polymorphism, as we see it, in that it suggests no obvious way of typing patterns of the form  $xy$  where  $x$  denotes an arbitrary piece of unstructured *data*. Additional differences with our work are:

- [13]: It does not introduce union types. No runtime matching error detection takes place (this is achieved via Progress in our paper).
- [11]: It deals with an untyped  $\rho$ -calculus. Hence no SR.
- [5, 12]: Neither union nor recursive types are considered.

**Structure of the paper.** Sec. 2 introduces the terms and operational semantics of CAP. The typing system is developed in Sec. 3 together with a precise definition of compatibility. Sec. 4 studies Safety: SR and Progress. Finally, we conclude. The document you are reading is the report including full proofs.

## 2 Syntax and Operational Semantics of CAP

We assume given an infinite set of term variables  $\mathbb{V}$  and constants  $\mathbb{C}$ . The syntax of CAP consists of four syntactic categories, namely **patterns**  $(p, q, \dots)$ , **terms**  $(s, t, \dots)$ , **data structures**  $(d, e, \dots)$  and **matchable forms**  $(m, n, \dots)$ :

$$\begin{array}{ll}
 p ::= x & \text{(matchable)} \quad t ::= x & \text{(variable)} \\
 | \quad \mathbf{c} & \text{(constant)} \quad | \quad \mathbf{c} & \text{(constant)} \\
 | \quad pp & \text{(compound)} \quad | \quad tt & \text{(application)} \\
 & | \quad p \rightarrow_{\theta} t \mid \dots \mid p \rightarrow_{\theta} t & \text{(abstraction)} \\
 \\
 d ::= \mathbf{c} & \text{(constant)} \quad m ::= d & \text{(data structure)} \\
 | \quad dt & \text{(compound)} \quad | \quad p \rightarrow_{\theta} t \mid \dots \mid p \rightarrow_{\theta} t & \text{(abstraction)}
 \end{array}$$

The set of patterns, terms, data structures and matchable forms are denoted  $\mathbb{P}$ ,  $\mathbb{T}$ ,  $\mathbb{D}$  and  $\mathbb{M}$ , resp. Variables occurring in patterns are called **matchables**. We often abbreviate  $p_1 \rightarrow_{\theta_1} s_1 \mid \dots \mid p_n \rightarrow_{\theta_n} s_n$  with  $(p_i \rightarrow_{\theta_i} s_i)_{i \in 1..n}$ . The  $\theta_i$  are typing contexts annotating the type assignments for the variables in  $p_i$  (cf. Sec. 3). The **free variables** of a term  $t$  (notation  $\text{fv}(t)$ ) are defined as expected; in a pattern  $p$  we call them **free matchables** ( $\text{fm}(p)$ ). All free matchables in each  $p_i$  are assumed to be bound in their respective bodies  $s_i$ . Positions in patterns and terms are defined as expected and denoted  $\pi, \pi', \dots$  ( $\epsilon$  denotes the root position). We write  $\text{pos}(s)$  for the set of positions of  $s$  and  $s|_{\pi}$  for the subterm of  $s$  occurring at position  $\pi$ .

A **substitution**  $(\sigma, \sigma_i, \dots)$  is a partial function from term variables to terms. If it assigns  $u_i$  to  $x_i$ ,  $i \in 1..n$ , then we write  $\{u_1/x_1, \dots, u_n/x_n\}$ . Its domain ( $\text{dom}(\sigma)$ )

is  $\{x_1, \dots, x_n\}$ . Also,  $\{\}$  is the identity substitution. We write  $\sigma s$  for the result of applying  $\sigma$  to term  $s$ . **Matchable forms** are required for defining the **matching operation**, described next.

Given a pattern  $p$  and a term  $s$ , the matching operation  $\{\{s/p\}\}$  determines whether  $s$  matches  $p$ . It may have one of three outcomes: success, fail (in which case it returns the special symbol **fail**) or undetermined (in which case it returns the special symbol **wait**). We say  $\{\{s/p\}\}$  is **decided** if it is either successful or it fails. In the former it yields a substitution  $\sigma$ ; in this case we write  $\{\{s/p\}\} = \sigma$ . The disjoint union of matching outcomes is given as follows (“ $\triangleq$ ” is used for definitional equality):

$$\begin{aligned} \text{fail} \uplus o &\triangleq \text{fail} & \text{wait} \uplus \sigma &\triangleq \text{wait} \\ o \uplus \text{fail} &\triangleq \text{fail} & \sigma \uplus \text{wait} &\triangleq \text{wait} \\ \sigma_1 \uplus \sigma_2 &\triangleq \sigma & \text{wait} \uplus \text{wait} &\triangleq \text{wait} \end{aligned}$$

where  $o$  denotes any possible output and  $\sigma_1 \uplus \sigma_2 \triangleq \sigma$  if the domains of  $\sigma_1$  and  $\sigma_2$  are disjoint. This always holds given that patterns are assumed to be linear (at most one occurrence of any matchable). The matching operation is defined as follows, where the defining clauses below are evaluated from top to bottom<sup>4</sup>:

$$\begin{aligned} \{\{u/x\}\} &\triangleq \{u/x\} \\ \{\{c/c\}\} &\triangleq \{\} \\ \{\{uv/pq\}\} &\triangleq \{\{u/p\}\} \uplus \{\{v/q\}\} && \text{if } uv \text{ is a matchable form} \\ \{\{u/p\}\} &\triangleq \text{fail} && \text{if } u \text{ is a matchable form} \\ \{\{u/p\}\} &\triangleq \text{wait} \end{aligned}$$

For example:  $\{\{x \rightarrow s/c\}\} = \text{fail}$ ;  $\{\{d/c\}\} = \text{fail}$ ;  $\{\{x/c\}\} = \text{wait}$  and  $\{\{xd/cc\}\} = \text{fail}$ . We now turn to the only reduction axiom of CAP:

$$\frac{\{\{u/p_i\}\} = \text{fail for all } i < j \quad \{\{u/p_j\}\} = \sigma_j \quad j \in 1..n}{(p_i \rightarrow_{\theta_i} s_i)_{i \in 1..n} u \rightarrow \sigma_j s_j} (\beta)$$

It may be applied under any context and states that if the argument  $u$  to an abstraction  $(p_i \rightarrow_{\theta_i} s_i)_{i \in 1..n}$  fails to match all patterns  $p_i$  with  $i < j$  and successfully matches pattern  $p_j$  (producing a substitution  $\sigma_j$ ), then the term  $(p_i \rightarrow_{\theta_i} s_i)_{i \in 1..n} u$  reduces to  $\sigma_j s_j$ .

The following example illustrates the use of the reduction rule and the matching operation:

$$\begin{aligned} &(\text{true} \rightarrow 1 \mid \text{false} \rightarrow 0)((\text{true} \rightarrow \text{false} \mid \text{false} \rightarrow \text{true})\text{true}) && (5) \\ &\rightarrow (\text{true} \rightarrow 1 \mid \text{false} \rightarrow 0)\{\{\text{true}/\text{true}\}\}\text{false} \\ &= (\text{true} \rightarrow 1 \mid \text{false} \rightarrow 0)\text{false} \\ &\rightarrow \{\{\text{false}/\text{false}\}\}0 && \{\{\text{false}/\text{true}\}\} = \text{fail} \\ &= 0 \end{aligned}$$

<sup>4</sup> This is simplification to the static patterns case of the matching operation introduced in [19].

**Proposition 2.1** *Reduction in CAP is confluent (CR).*

This result follows from a straightforward adaptation of the CR proof presented in [19] to our calculus. The key step is proving that the matching operation satisfies the *Rigid Matching Condition (RMC)* proposed in the cited work. Note that CAP is just the static patterns fragment of PPC where instead of the usual abstraction we have alternatives (*i.e.* we abstract multiple branches with the same constructor). Our contribution is on the typed variant of the calculus.

### 3 Typing System

This section presents  $\mu$ -types, the finite type expressions that shall be used for typing terms in CAP, their associated notions of equivalence and subtyping and then the typing schemes. Also, further examples and definitions associated to compatibility are included.

#### 3.1 Types

In order to ensure that patterns such as  $xy$  decompose only data structures rather than arbitrary terms, we shall introduce two sorts of typing expressions: *types* and *datatypes*, the latter being strictly included in the former. We assume given countably infinite sets  $\mathcal{V}_D$  of **datatype variables** ( $\alpha, \beta, \dots$ ),  $\mathcal{V}_A$  of **type variables** ( $X, Y, \dots$ ) and  $\mathcal{C}$  of **type constants** ( $\mathfrak{c}, \mathfrak{d}, \dots$ ). We define  $\mathcal{V} \triangleq \mathcal{V}_A \cup \mathcal{V}_D$  and use metavariables  $V, W, \dots$  to denote an arbitrary element in it. Likewise, we write  $a, b, \dots$  for elements in  $\mathcal{V} \cup \mathcal{C}$ . The sets  $\mathcal{T}_D$  of  $\mu$ -**datatypes** and  $\mathcal{T}$  of  $\mu$ -**types**, resp., are inductively defined as follows:

|                |                     |               |                    |
|----------------|---------------------|---------------|--------------------|
| $D ::= \alpha$ | (datatype variable) | $A ::= X$     | (type variable)    |
| $\mathfrak{c}$ | (atom)              | $D$           | (datatype)         |
| $D @ A$        | (compound)          | $A \supset A$ | (type abstraction) |
| $D \oplus D$   | (union)             | $A \oplus A$  | (union)            |
| $\mu\alpha.D$  | (recursion)         | $\mu X.A$     | (recursion)        |

**Remark 3.1** A type of the form  $\mu\alpha.A$  is not valid in general since it may produce invalid unfoldings. For example,  $\mu\alpha.\alpha \supset \alpha = (\mu\alpha.\alpha \supset \alpha) \supset (\mu\alpha.\alpha \supset \alpha)$ . On the other hand, types of the form  $\mu X.D$  are not necessary since they denote the solution to the equation  $X = D$ , hence  $X$  is a variable representing a datatype.

We consider  $\oplus$  to bind tighter than  $\supset$ , while  $@$  binds tighter than  $\oplus$ . Therefore  $D @ A \oplus A' \supset B$  means  $((D @ A) \oplus A') \supset B$ . Additionally, when referring to a finite series of consecutive unions such as  $A_1 \oplus \dots \oplus A_n$  we will use the simplified notation  $\bigoplus_{i \in 1..n} A_i$ . This notation is not strict on how subexpressions  $A_i$  are associated hence, in principle, it refers to any of all possible associations. In the next section we present an equivalence relation on  $\mu$ -types that will identify all these associations. We often write  $\mu V.A$  to mean either  $\mu\alpha.D$  or  $\mu X.A$ . A **non-union  $\mu$ -type**  $A$  is a  $\mu$ -type of one of the following forms:  $\alpha, \mathfrak{c}, D @ A, X, A \supset B$  or  $\mu V.A$  with  $A$  a non-union



$$\begin{array}{c}
 \frac{}{\vdash A \simeq_\mu A} \text{(E-REFL)} \quad \frac{\vdash A \simeq_\mu B \quad \vdash B \simeq_\mu C}{\vdash A \simeq_\mu C} \text{(E-TRANS)} \quad \frac{\vdash A \simeq_\mu B}{\vdash B \simeq_\mu A} \text{(E-SYMM)} \\
 \\
 \frac{\vdash A \simeq_\mu A' \quad \vdash B \simeq_\mu B'}{\vdash A \supset B \simeq_\mu A' \supset B'} \text{(E-FUNC)} \quad \frac{\vdash D \simeq_\mu D' \quad \vdash A \simeq_\mu A'}{\vdash D @ A \simeq_\mu D' @ A'} \text{(E-COMP)} \\
 \\
 \frac{}{\vdash A \oplus A \simeq_\mu A} \text{(E-UNION-IDEM)} \quad \frac{}{\vdash A \oplus B \simeq_\mu B \oplus A} \text{(E-UNION-COMM)} \\
 \\
 \frac{}{\vdash A \oplus (B \oplus C) \simeq_\mu (A \oplus B) \oplus C} \text{(E-UNION-ASSOC)} \\
 \\
 \frac{\vdash A \simeq_\mu A' \quad \vdash B \simeq_\mu B'}{\vdash A \oplus B \simeq_\mu A' \oplus B'} \text{(E-UNION)} \quad \frac{\vdash A \simeq_\mu B}{\vdash \mu V.A \simeq_\mu \mu V.B} \text{(E-REC)} \\
 \\
 \frac{}{\vdash \mu V.A \simeq_\mu \{\mu V.A/V\} A} \text{(E-FOLD)} \quad \frac{\vdash A \simeq_\mu \{A/V\} B \quad \mu V.B \text{ contractive}}{\vdash A \simeq_\mu \mu V.B} \text{(E-CONTR)}
 \end{array}$$

 Fig. 1. Type equivalence for  $\mu$ -types

$\mu$ -type. We assume  $\mu$ -types are **contractive**:  $\mu V.A$  is contractive if  $V$  occurs in  $A$  only under a type constructor  $\supset$  or  $@$ , if at all. We henceforth redefine  $\mathcal{T}$  to be the set of **contractive  $\mu$ -types**.  $\mu$ -types come equipped with a notion of equivalence  $\simeq_\mu$  and subtyping  $\preceq_\mu$ .

**Definition 3.2** (i)  $\simeq_\mu$  is defined by the schemes in Fig. 1.

(ii)  $\preceq_\mu$  is defined by the schemes in Fig. 2 where a subtyping context  $\Sigma$  is a set of assumptions over type variables of the form  $V \preceq_\mu W$  with  $V, W \in \mathcal{V}$ .

(E-REC) actually encodes two rules, one for datatypes ( $\mu\alpha.D$ ) and one for arbitrary types ( $\mu X.A$ ). Likewise for (E-FOLD) and (E-CONTR). The relation resulting from dropping (E-CONTR) [3, 7] is called weak type equivalence [9] and is known to be too weak to capture equivalence of its coinductive formulation (required for our proof of invertibility of subtyping *cf.* Prop. 3.32); for example, types  $\mu X.A \supset A \supset X$  and  $\mu X.A \supset X$  cannot be equated.

Regarding the subtyping rules, we adopt those for union of [27]. It should be noted that the naïve variant of (S-REC) in which  $\Sigma \vdash \mu V.A \preceq_\mu \mu V.B$  is deduced from  $\Sigma \vdash A \preceq_\mu B$ , is known to be unsound [1]. We often abbreviate  $\vdash A \preceq_\mu B$  as  $A \preceq_\mu B$ .

We can now use notation  $\bigoplus_{i \in I} A_i$  on contractive  $\mu$ -types to denote several consecutive applications of the binary operator  $\oplus$  irrespective of how they are associated. All such associations yield equivalent  $\mu$ -types. Such expressions will be useful to prove the correspondence between the types as trees formulation and the contractive  $\mu$ -types of the current section. To that end we introduce the following lemmas



$$\begin{array}{c}
 \frac{}{\Sigma \vdash A \preceq_\mu A} \text{ (S-REFL)} \quad \frac{}{\Sigma, V \preceq_\mu W \vdash V \preceq_\mu W} \text{ (S-HYP)} \quad \frac{\vdash A \simeq_\mu B}{\Sigma \vdash A \preceq_\mu B} \text{ (S-EQ)} \\
 \\
 \frac{\Sigma \vdash A \preceq_\mu B \quad \Sigma \vdash B \preceq_\mu C}{\Sigma \vdash A \preceq_\mu C} \text{ (S-TRANS)} \quad \frac{\Sigma \vdash D \preceq_\mu D' \quad \Sigma \vdash A \preceq_\mu A'}{\Sigma \vdash D @ A \preceq_\mu D' @ A'} \text{ (S-COMP)} \\
 \\
 \frac{\Sigma \vdash A \preceq_\mu A' \quad \Sigma \vdash B \preceq_\mu B'}{\Sigma \vdash A' \supset B \preceq_\mu A \supset B'} \text{ (S-FUNC)} \quad \frac{\Sigma \vdash A \preceq_\mu C \quad \Sigma \vdash B \preceq_\mu C}{\Sigma \vdash A \oplus B \preceq_\mu C} \text{ (S-UNION-L)} \\
 \\
 \frac{\Sigma \vdash A \preceq_\mu B}{\Sigma \vdash A \preceq_\mu B \oplus C} \text{ (S-UNION-R1)} \quad \frac{\Sigma \vdash A \preceq_\mu C}{\Sigma \vdash A \preceq_\mu B \oplus C} \text{ (S-UNION-R2)} \\
 \\
 \frac{\Sigma, V \preceq_\mu W \vdash A \preceq_\mu B \quad W \notin \text{fv}(A) \quad V \notin \text{fv}(B)}{\Sigma \vdash \mu V. A \preceq_\mu \mu W. B} \text{ (S-REC)}
 \end{array}$$

 Fig. 2. Strong subtyping for  $\mu$ -types

that extend the associative, commutative and idempotent properties to arbitrary unions.

To simplify the presentation of the proofs, we often resort to the following reasoning (or its symmetric variant)

$$\frac{\frac{\vdots}{A \simeq_\mu B} X \quad \frac{}{C \simeq_\mu C} \text{ (E-REFL)}}{A \oplus C \simeq_\mu B \oplus C} \text{ (E-UNION)}$$

by only stating (X) (*i.e.* a rule, lemma, inductive hypothesis, etc.). Thus, we say that  $A \oplus C \simeq_\mu B \oplus C$  by (X) or, in other words, apply (X) within a *union context*.

**Lemma 3.3** *Let  $A$  and  $A'$  be two distinct associations of  $\bigoplus_{i \in 1..n} A_i$ . Then,  $A \simeq_\mu A'$ .*

**Proof.** Direct consequence of (E-UNION-ASSOC). □

**Lemma 3.4** *Let  $p$  be a permutation over  $1..n$ . Then,  $\bigoplus_{i \in 1..n} A_i \simeq_\mu \bigoplus_{i \in 1..n} A_{p(i)}$ .*

**Proof.** By induction on  $n$ .

- $n = 1$ . This case is immediate since  $p = id$ .

- $n > 1$ . Without loss of generality we can consider  $p$  to be the function

$$p(i) = \begin{cases} p'(i) & \text{if } i < k \\ n & \text{if } i = k \\ p'(i-1) & \text{if } i > k \end{cases}$$

where  $p'$  is a permutation over  $1..n-1$  and  $k \in 1..n$ . That is,  $p$  permutes  $k$  with  $n$  and behaves like  $p'$  on every other position. Then,

$$\begin{aligned} \bigoplus_{i \in 1..n} A_i &\simeq_{\mu} (\bigoplus_{i \in 1..n-1} A_i) \oplus A_n && \text{Lem. 3.3} \\ &\simeq_{\mu} (\bigoplus_{i \in 1..n-1} A_{p'(i)}) \oplus A_n && \text{by IH} \end{aligned}$$

If  $k = n$  we are done, since  $(\bigoplus_{i \in 1..n-1} A_{p'(i)}) \oplus A_n \simeq_{\mu} \bigoplus_{i \in 1..n} A_{p(i)}$  by Lem. 3.3. If not (*i.e.*  $k \in 1..n-1$ ) we just need to apply (E-UNION-COMM) to the proper subexpression

$$\begin{aligned} \bigoplus_{i \in 1..n} A_i &\simeq_{\mu} (\bigoplus_{i \in 1..n-1} A_{p'(i)}) \oplus A_n \\ &\simeq_{\mu} (\bigoplus_{i \in 1..k-1} A_{p'(i)}) \oplus ((\bigoplus_{i \in k..n-1} A_{p'(i)}) \oplus A_n) && \text{Lem. 3.3} \\ &\simeq_{\mu} (\bigoplus_{i \in 1..k-1} A_{p'(i)}) \oplus (A_n \oplus (\bigoplus_{i \in k..n-1} A_{p'(i)})) && \text{(E-UNION-COMM)} \\ &\simeq_{\mu} \bigoplus_{i \in 1..n} A_{p(i)} && \text{Lem. 3.3} \end{aligned}$$

□

**Lemma 3.5** *Let  $J_m = \langle J, m \rangle$  be a finite multiset<sup>5</sup> such that  $J \subseteq 1..n$ , then  $\bigoplus_{i \in 1..n} A_i \simeq_{\mu} (\bigoplus_{i \in 1..n} A_i) \oplus (\bigoplus_{j \in J_m} A_j)$ .*

**Proof.** This proof is by induction on  $\#(J_m)$  (the cardinality of the multiset  $J_m$ ).

- $\#(J_m) = 0$ . This case is immediate by Lem. 3.3 (note that both sides of the equivalence may be associated differently, thus (E-REFL) is not enough).
- $\#(J_m) > 0$ . Let  $k \in J_m$ . Then

$$\begin{aligned} (\bigoplus_{i \in 1..n} A_i) \oplus (\bigoplus_{j \in J_m} A_j) &\simeq_{\mu} ((\bigoplus_{i \in 1..n} A_i) \oplus (\bigoplus_{j \in (J_m \setminus \{k\})} A_j)) \oplus A_k && \text{Lem. 3.4} \\ &\simeq_{\mu} (\bigoplus_{i \in 1..n} A_i) \oplus A_k && \text{by IH} \\ &\simeq_{\mu} (\bigoplus_{i \in 1..n, i \neq k} A_i) \oplus (A_k \oplus A_k) && \text{Lem. 3.4} \\ &\simeq_{\mu} (\bigoplus_{i \in 1..n, i \neq k} A_i) \oplus A_k && \text{(E-UNION-IDEM)} \\ &\simeq_{\mu} \bigoplus_{i \in 1..n} A_i && \text{Lem. 3.4} \end{aligned}$$

□

The following lemma presents an admissible rule regarding union types that shall be used later to relate  $\simeq_{\mu}$  with its coinductive characterisation. Note that in this

<sup>5</sup> Recall that a *multiset* is a pair  $\mathcal{M} = \langle \mathcal{X}, m \rangle$  where  $\mathcal{X}$  is the underlying set of  $\mathcal{M}$  and  $m : \mathcal{X} \rightarrow \mathbb{N}$  is its *multiplicity function*. We will usually denote  $\mathcal{M}$  with  $\mathcal{X}$  when there is no ambiguity or the meaning is clear from the context.

case there is no need for types  $A_i, B_j$  to be non-union types below.

**Lemma 3.6** *Let  $A = \bigoplus_{i \in 1..n} A_i, B = \bigoplus_{j \in 1..m} B_j$  and  $f : 1..n \rightarrow 1..m, g : 1..m \rightarrow 1..n$  functions such that  $A_i \simeq_\mu B_{f(i)}$  and  $A_{g(j)} \simeq_\mu B_j$  for every  $i \in 1..n, j \in 1..m$ . Then,  $\bigoplus_{i \in 1..n} A_i \simeq_\mu \bigoplus_{j \in 1..m} B_j$ .*

**Proof.** It is immediate to see that for every multiset of indexes  $I \subseteq 1..n, \bigoplus_{i \in I} A_i \simeq_\mu \bigoplus_{i \in I} B_{f(i)}$ , by applying (E-UNION) as many times as needed and resorting to Lem. 3.3 if necessary. Similarly,  $\bigoplus_{j \in J} B_j \simeq_\mu \bigoplus_{j \in J} A_{g(j)}$  for  $J \subseteq 1..m$ . So let's consider some multisets and see how they relate to each other to finish our analysis

$$\begin{aligned} I &\triangleq \{i \mid i \in 1..n, i \in \text{img}(g)\} \\ I' &\triangleq \{i \mid i \in 1..n, i \notin \text{img}(g)\} \\ G &\triangleq \{g(j) \mid j \in 1..m\} \\ F &\triangleq \{f(i) \mid i \in 1..n, i \notin \text{img}(g)\} \end{aligned}$$

First notice that, by definition,  $I$  and  $I'$  have no repeated elements and

$$G = I \cup G' \quad \text{with} \quad G' \subseteq I \tag{6}$$

where  $G'$  simply holds the repeated elements of  $G$ . Additionally we have

$$F \subseteq 1..m \tag{7}$$

Finally, we can conclude by resorting to some previous results

$$\begin{aligned} A &= \bigoplus_{i \in 1..n} A_i \\ &\simeq_\mu (\bigoplus_{i \in I} A_i) \oplus (\bigoplus_{i \in I'} A_i) && \text{Lem. 3.4} \\ &\simeq_\mu ((\bigoplus_{i \in I} A_i) \oplus (\bigoplus_{i \in G'} A_i)) \oplus (\bigoplus_{i \in I'} A_i) && \text{Lem. 3.5 with (6)} \\ &= (\bigoplus_{i \in G} A_i) \oplus (\bigoplus_{i \in I'} A_i) \\ &= (\bigoplus_{j \in 1..m} A_{g(j)}) \oplus (\bigoplus_{i \in I'} A_i) \\ &\simeq_\mu (\bigoplus_{j \in 1..m} B_j) \oplus (\bigoplus_{i \in I'} A_i) \\ &= (\bigoplus_{j \in 1..m} B_j) \oplus (\bigoplus_{\substack{i \in 1..n \\ i \notin \text{img}(g)}} A_i) \\ &\simeq_\mu (\bigoplus_{j \in 1..m} B_j) \oplus (\bigoplus_{\substack{i \in 1..n \\ i \notin \text{img}(g)}} B_{f(i)}) \\ &= (\bigoplus_{j \in 1..m} B_j) \oplus (\bigoplus_{j \in F} B_j) \\ &\simeq_\mu \bigoplus_{j \in 1..m} B_j && \text{Lem. 3.5 with (7)} \\ &= B \end{aligned}$$

□

$$\begin{array}{c}
 \frac{}{a \simeq_{\mathfrak{T}} a} \text{ (E-REFL-T)} \\
 \\
 \frac{\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{A}' \quad \mathcal{B} \simeq_{\mathfrak{T}} \mathcal{B}'}{\mathcal{A} \supset \mathcal{B} \simeq_{\mathfrak{T}} \mathcal{A}' \supset \mathcal{B}'} \text{ (E-FUNC-T)} \qquad \frac{\mathcal{D} \simeq_{\mathfrak{T}} \mathcal{D}' \quad \mathcal{A} \simeq_{\mathfrak{T}} \mathcal{A}'}{\mathcal{D} @ \mathcal{A} \simeq_{\mathfrak{T}} \mathcal{D}' @ \mathcal{A}'} \text{ (E-COMP-T)} \\
 \\
 \frac{\mathcal{A}_i \simeq_{\mathfrak{T}} \mathcal{B}_{f(i)} \quad f : 1..n \rightarrow 1..m \quad \mathcal{A}_i, \mathcal{B}_j \neq \oplus \quad n + m > 2 \quad \mathcal{A}_{g(j)} \simeq_{\mathfrak{T}} \mathcal{B}_j \quad g : 1..m \rightarrow 1..n}{\bigoplus_{i \in 1..n} \mathcal{A}_i \simeq_{\mathfrak{T}} \bigoplus_{j \in 1..m} \mathcal{B}_j} \text{ (E-UNION-T)}
 \end{array}$$

Fig. 3. Equivalence relation for infinite types

### 3.1.1 Types as trees

Type safety, addressed in the Sec.4, also relies on  $\preceq_{\mu}$  enjoying the fundamental property of *invertibility* of non-union types (cf. Prop. 3.32):

- (i) If  $D @ A \preceq_{\mu} D' @ A'$ , then  $D \preceq_{\mu} D'$  and  $A \preceq_{\mu} A'$ .
- (ii) If  $A \supset B \preceq_{\mu} A' \supset B'$ , then  $A' \preceq_{\mu} A$  and  $B \preceq_{\mu} B'$ .

To prove this we appeal to the standard tree interpretation of terms and formulate an equivalent coinductive definition of equivalence and subtyping ( $\preceq_{\mathfrak{T}}$ ). For the latter, invertibility of non-union types is proved coinductively, (Lem. 3.17), entailing Prop. 3.32.

Consider **type constructors**  $@$  and  $\supset$  together with **type connector**  $\oplus$  and the ranked alphabet  $\mathfrak{L} \triangleq \{a^0 \mid a \in \mathcal{V} \cup \mathcal{C}\} \cup \{a^2, \supset^2, \oplus^2\}$ . We write  $\mathfrak{T}$  for the set of (possibly) **infinite types** with symbols in  $\mathfrak{L}$ . This is a standard construction [6, 16] given by the metric completion based on a simple depth function measuring the distance from the root to the minimum conflicting node in two trees. Perhaps worth mentioning is that the type connector  $\oplus$  does not contribute to the depth (hence the reason for calling it a connector rather than a constructor) excluding types consisting of infinite branches of  $\oplus$ , such as  $(\dots \oplus \dots) \oplus (\dots \oplus \dots)$ , from  $\mathfrak{T}$ . We use meta-variables  $\mathcal{A}, \mathcal{B}, \dots$  to denote elements of  $\mathfrak{T}$ .

**Remark 3.7** For any  $\star \in \mathfrak{L}$ , we write  $\mathcal{A} \neq \star$  to mean that  $\mathcal{A}(\epsilon) \neq \star$ ,  $\epsilon$  being the root position of the tree. For example,  $\mathcal{A} \neq \oplus$  means that  $\mathcal{A}$  is a *non-union type*. Any type  $\mathcal{A}$  can be written as  $\mathcal{A} = \bigoplus_{i \in 1..n} \mathcal{A}_i$  (dubbed a *maximal union type*) where  $\mathcal{A}_i \neq \oplus$  for all  $i \in 1..n$  with  $n \in \mathbb{N}$ , irrespective of how their arguments are associated. All such associations yield equivalent infinite types in a sense to be made precise shortly.

### 3.1.2 Equivalence of Infinite Types

**Definition 3.8** Infinite type equivalence, written  $\simeq_{\mathfrak{T}}$ , is defined by the coinductive interpretation of the schemes of Fig. 3.

Note that (E-UNION-T) is actually a rule scheme, representing all possible associations within maximal union types  $\mathcal{A} = \bigoplus_{i \in 1..n} \mathcal{A}_i$  and  $\mathcal{B} = \bigoplus_{j \in 1..m} \mathcal{B}_j$ . Each

instance of the rule states that every  $A_i$  must be equivalent to some  $B_j$  via a function  $f : 1..n \rightarrow 1..m$  and vice versa (with  $g : 1..m \rightarrow 1..n$ ). Note that the type connector  $\oplus$  is seen to be not only associative and commutative but also idempotent.

Formally, let  $\Phi_{\simeq_{\mathfrak{T}}} : \wp(\mathfrak{T} \times \mathfrak{T}) \rightarrow \wp(\mathfrak{T} \times \mathfrak{T})$  be the functional associated to the rules in Fig. 3, defined as follows:

$$\begin{aligned} \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S}) = & \{ \langle a, a \rangle \mid a \in \mathcal{V} \cup \mathcal{C} \} \\ & \cup \{ \langle \mathcal{D} @ \mathcal{A}, \mathcal{D}' @ \mathcal{A}' \rangle \mid \langle \mathcal{D}, \mathcal{D}' \rangle, \langle \mathcal{A}, \mathcal{A}' \rangle \in \mathcal{S} \} \\ & \cup \{ \langle \mathcal{A} \supset \mathcal{B}, \mathcal{A}' \supset \mathcal{B}' \rangle \mid \langle \mathcal{A}, \mathcal{A}' \rangle, \langle \mathcal{B}, \mathcal{B}' \rangle \in \mathcal{S} \} \\ & \cup \{ \langle \bigoplus_{i \in 1..n} \mathcal{A}_i, \bigoplus_{j \in 1..m} \mathcal{B}_j \rangle \mid \mathcal{A}_i, \mathcal{B}_j \neq \oplus, n + m > 2 \\ & \quad \exists f : 1..n \rightarrow 1..m \text{ s.t. } \langle \mathcal{A}_i, \mathcal{B}_{f(i)} \rangle \in \mathcal{S}, \\ & \quad \exists g : 1..m \rightarrow 1..n \text{ s.t. } \langle \mathcal{A}_{g(j)}, \mathcal{B}_j \rangle \in \mathcal{S} \} \end{aligned}$$

Then  $\simeq_{\mathfrak{T}} \triangleq \nu \Phi_{\simeq_{\mathfrak{T}}}$ . Now we show that it is indeed an equivalence relation.

**Lemma 3.9**  $\simeq_{\mathfrak{T}}$  is an equivalence relation (i.e. reflexive, symmetric and transitive).

**Proof.** The three properties are proved by showing that the sets defining them are  $\Phi_{\simeq_{\mathfrak{T}}}$ -dense. Then we conclude by the coinductive principle<sup>6</sup> that the properties hold on  $\simeq_{\mathfrak{T}}$ .

- Reflexivity:  $Refl \triangleq \{ \langle \mathcal{A}, \mathcal{A} \rangle \mid \mathcal{A} \in \mathfrak{T} \}$ . Let  $\langle \mathcal{A}, \mathcal{A} \rangle \in Refl$ . We proceed by analyzing the shape of  $\mathcal{A}$ :
  - $\mathcal{A} = a$ . Immediate since  $\langle a, a \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(Refl)$  for every  $a \in \mathcal{V} \cup \mathcal{C}$ .
  - $\mathcal{A} = \mathcal{D} @ \mathcal{A}'$ . By definition of reflexivity  $\langle \mathcal{D}, \mathcal{D} \rangle, \langle \mathcal{A}', \mathcal{A}' \rangle \in Refl$ . Then  $\langle \mathcal{A}, \mathcal{A} \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(Refl)$ .
  - $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$ . Similarly to the previous case, we have  $\langle \mathcal{A}', \mathcal{A}' \rangle, \langle \mathcal{A}'', \mathcal{A}'' \rangle \in Refl$ . Hence  $\langle \mathcal{A}, \mathcal{A} \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(Refl)$ .
  - $\mathcal{A} = \bigoplus_{i \in 1..n} \mathcal{A}_i$  with  $\mathcal{A}_i \neq \oplus$  for  $i \in 1..n, n > 1$ . Then, since  $\langle \mathcal{A}_i, \mathcal{A}_i \rangle \in Refl$  and  $n + n > 2$ , we conclude  $\langle \bigoplus_{i \in 1..n} \mathcal{A}_i, \bigoplus_{i \in 1..n} \mathcal{A}_i \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(Refl)$  by considering  $f = g = id$  (the identity function).
- Symmetry:  $Symm(\mathcal{S}) \triangleq \{ \langle \mathcal{B}, \mathcal{A} \rangle \mid \langle \mathcal{A}, \mathcal{B} \rangle \in \mathcal{S} \}$ . We show that  $Symm(\simeq_{\mathfrak{T}}) \subseteq \simeq_{\mathfrak{T}}$ .

Let  $\langle \mathcal{A}, \mathcal{B} \rangle \in Symm(\simeq_{\mathfrak{T}})$ , then  $\langle \mathcal{B}, \mathcal{A} \rangle \in \simeq_{\mathfrak{T}} = \Phi_{\simeq_{\mathfrak{T}}}(\simeq_{\mathfrak{T}})$ . By Rem. 3.7 we can consider maximal union types

$$\begin{aligned} \mathcal{A} &= \bigoplus_{i \in 1..n} \mathcal{A}_i \quad \text{with} \quad \mathcal{A}_i \neq \oplus, i \in 1..n \\ \mathcal{B} &= \bigoplus_{j \in 1..m} \mathcal{B}_j \quad \text{with} \quad \mathcal{B}_j \neq \oplus, j \in 1..m \end{aligned}$$

and we have two separate cases to analyze:

- (i) If  $n = m = 1$ , then both  $\mathcal{A}$  and  $\mathcal{B}$  are non-union types. Now we proceed by analyzing the shape of  $\mathcal{B}$ :

<sup>6</sup> Coinductive principle: if  $\mathcal{X}$  is  $\Phi$ -dense, then  $\mathcal{X} \subseteq \nu \Phi$ .

- $\mathcal{B} = a$ . Then  $\mathcal{A} = a$  by definition of  $\Phi_{\simeq_{\mathfrak{T}}}$  and the result is immediate since  $\langle a, a \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\text{Symm}(\simeq_{\mathfrak{T}}))$  for every  $a \in \mathcal{V} \cup \mathcal{C}$ .
  - $\mathcal{B} = \mathcal{D}' @ \mathcal{B}'$ . Again, by definition, we have  $\mathcal{A} = \mathcal{D} @ \mathcal{A}'$  with  $\langle \mathcal{D}', \mathcal{D} \rangle, \langle \mathcal{B}', \mathcal{A}' \rangle \in \simeq_{\mathfrak{T}}$ . Then  $\langle \mathcal{D}, \mathcal{D}' \rangle, \langle \mathcal{A}', \mathcal{B}' \rangle \in \text{Symm}(\simeq_{\mathfrak{T}})$  and we conclude  $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\text{Symm}(\simeq_{\mathfrak{T}}))$ .
  - $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$ . Similarly,  $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$  with  $\langle \mathcal{B}', \mathcal{A}' \rangle, \langle \mathcal{B}'', \mathcal{A}'' \rangle \in \simeq_{\mathfrak{T}}$ . Hence  $\langle \mathcal{A}', \mathcal{B}' \rangle, \langle \mathcal{A}'', \mathcal{B}'' \rangle \in \text{Symm}(\simeq_{\mathfrak{T}})$  and we conclude  $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\text{Symm}(\simeq_{\mathfrak{T}}))$ .
- (ii) If not, we have  $n + m > 2$  and only the rule (E-UNION-T) applies. Then

$$\begin{aligned} \exists g : 1..m \rightarrow 1..n \quad \text{s.t.} \quad \langle \mathcal{B}_j, \mathcal{A}_{g(j)} \rangle &\in \simeq_{\mathfrak{T}} \quad \text{for every } j \in 1..m \\ \exists f : 1..n \rightarrow 1..m \quad \text{s.t.} \quad \langle \mathcal{B}_{f(i)}, \mathcal{A}_i \rangle &\in \simeq_{\mathfrak{T}} \quad \text{for every } i \in 1..n \end{aligned}$$

Applying symmetry we get  $\langle \mathcal{A}_i, \mathcal{B}_{f(i)} \rangle, \langle \mathcal{A}_{g(j)}, \mathcal{B}_j \rangle \in \text{Symm}(\simeq_{\mathfrak{T}})$  for every  $i \in 1..n, j \in 1..m$ . Thus, we conclude  $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\text{Symm}(\simeq_{\mathfrak{T}}))$ .

- Transitivity:  $\text{Trans}(\mathcal{S}) \triangleq \{ \langle \mathcal{A}, \mathcal{B} \rangle \mid \exists \mathcal{C} \in \mathfrak{T}. \langle \mathcal{A}, \mathcal{C} \rangle, \langle \mathcal{C}, \mathcal{B} \rangle \in \mathcal{S} \}$ . As before, we show that  $\text{Trans}(\simeq_{\mathfrak{T}}) \subseteq \simeq_{\mathfrak{T}}$ . Let  $\langle \mathcal{A}, \mathcal{B} \rangle \in \text{Trans}(\simeq_{\mathfrak{T}})$ , then there exists  $\mathcal{C} \in \mathfrak{T}$  such that  $\langle \mathcal{A}, \mathcal{C} \rangle, \langle \mathcal{C}, \mathcal{B} \rangle \in \simeq_{\mathfrak{T}} = \Phi_{\simeq_{\mathfrak{T}}}(\simeq_{\mathfrak{T}})$ . Again, we resort to Rem. 3.7 and consider maximal union types

$$\begin{aligned} \mathcal{A} &= \bigoplus_{i \in 1..n} \mathcal{A}_i \quad \text{with} \quad \mathcal{A}_i \neq \oplus, i \in 1..n \\ \mathcal{B} &= \bigoplus_{j \in 1..m} \mathcal{B}_j \quad \text{with} \quad \mathcal{B}_j \neq \oplus, j \in 1..m \\ \mathcal{C} &= \bigoplus_{k \in 1..l} \mathcal{C}_k \quad \text{with} \quad \mathcal{C}_k \neq \oplus, k \in 1..l \end{aligned}$$

- (i) If  $n = m = l = 1$  (i.e. all three are non-union types), we proceed by analyzing the shape of  $\mathcal{C}$ :
- $\mathcal{C} = a$ . By definition of  $\Phi_{\simeq_{\mathfrak{T}}}$ ,  $\mathcal{A} = a$  and  $\mathcal{B} = a$ . Then  $\langle \mathcal{A}, \mathcal{B} \rangle = \langle a, a \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\text{Trans}(\simeq_{\mathfrak{T}}))$ .
  - $\mathcal{C} = \mathcal{D}'' @ \mathcal{C}'$ . Once again by definition of  $\Phi_{\simeq_{\mathfrak{T}}}$ ,  $\mathcal{A} = \mathcal{D} @ \mathcal{A}'$  with  $\langle \mathcal{D}, \mathcal{D}'' \rangle, \langle \mathcal{A}', \mathcal{C}' \rangle \in \simeq_{\mathfrak{T}}$  and  $\mathcal{B} = \mathcal{D}' @ \mathcal{B}'$  with  $\langle \mathcal{D}'', \mathcal{D}' \rangle, \langle \mathcal{C}', \mathcal{B}' \rangle \in \simeq_{\mathfrak{T}}$ . Then  $\langle \mathcal{D}, \mathcal{D}' \rangle, \langle \mathcal{A}', \mathcal{B}' \rangle \in \text{Trans}(\simeq_{\mathfrak{T}})$  and we conclude  $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\text{Trans}(\simeq_{\mathfrak{T}}))$ .
  - $\mathcal{C} = \mathcal{C}' \supset \mathcal{C}''$ . Similarly, we have  $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$  and  $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$  with  $\langle \mathcal{A}', \mathcal{C}' \rangle, \langle \mathcal{A}'', \mathcal{C}'' \rangle, \langle \mathcal{C}', \mathcal{B}' \rangle, \langle \mathcal{C}'', \mathcal{B}'' \rangle \in \simeq_{\mathfrak{T}}$ . By transitivity  $\langle \mathcal{A}', \mathcal{B}' \rangle, \langle \mathcal{A}'', \mathcal{B}'' \rangle \in \text{Trans}(\simeq_{\mathfrak{T}})$  and  $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\text{Trans}(\simeq_{\mathfrak{T}}))$ .
- (ii) If not (i.e.  $n + m + l > 3$ ), we have three different situations to consider:
- (i)  $n + l > 2$  and  $m + l > 2$ ; (ii)  $n > 1$  and  $m = l = 1$ ; or (iii)  $m > 1$  and  $n = l = 1$ . In terms of applied rules to derive  $\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{C}$  and  $\mathcal{C} \simeq_{\mathfrak{T}} \mathcal{B}$ , in the former case the only possibility is (E-UNION-T) on both sides, while in the latter two we have (E-UNION-T) on one side and any of the other three rules ((E-REFL-T), (E-COMP-T), (E-FUNC-T)) on the other. Note that this last two cases are symmetric, therefore we only analyse cases (i) and (ii) below:

(i)  $n + l > 2$  and  $m + l > 2$ . By definition of  $\Phi_{\simeq_{\mathfrak{T}}}$

$$\begin{aligned} \exists f : 1..n \rightarrow 1..l \quad \text{s.t.} \quad \langle \mathcal{A}_i, \mathcal{C}_{f(i)} \rangle &\in \simeq_{\mathfrak{T}} \quad \text{for every } i \in 1..n \\ \exists g : 1..l \rightarrow 1..n \quad \text{s.t.} \quad \langle \mathcal{A}_{g(k)}, \mathcal{C}_k \rangle &\in \simeq_{\mathfrak{T}} \quad \text{for every } k \in 1..l \\ \exists f' : 1..l \rightarrow 1..m \quad \text{s.t.} \quad \langle \mathcal{C}_k, \mathcal{B}_{f'(k)} \rangle &\in \simeq_{\mathfrak{T}} \quad \text{for every } k \in 1..l \\ \exists g' : 1..m \rightarrow 1..l \quad \text{s.t.} \quad \langle \mathcal{C}_{g'(j)}, \mathcal{B}_j \rangle &\in \simeq_{\mathfrak{T}} \quad \text{for every } j \in 1..m \end{aligned}$$

Then, we have  $\langle \mathcal{A}_i, \mathcal{C}_{f(i)} \rangle, \langle \mathcal{C}_{f(i)}, \mathcal{B}_{f'(f(i))} \rangle \in \simeq_{\mathfrak{T}}$  for every  $i \in 1..n$ , and  $\langle \mathcal{A}_{g(g'(j))}, \mathcal{C}_{g'(j)} \rangle, \langle \mathcal{C}_{g'(j)}, \mathcal{B}_j \rangle \in \simeq_{\mathfrak{T}}$  for every  $j \in 1..m$ .

Here we have two possible situations. If  $n = m = 1$  (hence  $l > 1$ ) it is necessarily the case  $\langle \mathcal{A}, \mathcal{C}_{f(1)} \rangle, \langle \mathcal{C}_{f(1)}, \mathcal{B} \rangle \in \simeq_{\mathfrak{T}}$  with all three non-union types. Then we can safely conclude, by the previous analysis made in case 1, that  $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\text{Trans}(\simeq_{\mathfrak{T}}))$ .

If not (i.e.  $n + m > 2$ ), taking  $f'' = f' \circ f : 1..n \rightarrow 1..m$  we get  $\langle \mathcal{A}_i, \mathcal{B}_{f''(i)} \rangle \in \text{Trans}(\simeq_{\mathfrak{T}})$ . Similarly,  $\langle \mathcal{A}_{g''(j)}, \mathcal{B}_j \rangle \in \text{Trans}(\simeq_{\mathfrak{T}})$  for every  $j \in 1..m$  with  $g'' = g \circ g' : 1..m \rightarrow 1..n$ . Finally we conclude by (E-UNION-T),  $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\text{Trans}(\simeq_{\mathfrak{T}}))$ .

(ii)  $n > 1$  and  $m = l = 1$ . Then, by definition of  $\Phi_{\simeq_{\mathfrak{T}}}$ ,  $f : 1..n \rightarrow 1$  is a constant function and we have  $\langle \mathcal{A}_i, \mathcal{C} \rangle \in \simeq_{\mathfrak{T}}$  for every  $i \in 1..n$ . On the other hand  $\langle \mathcal{C}, \mathcal{B} \rangle \in \simeq_{\mathfrak{T}}$  by hypothesis. By transitivity once again we get  $\langle \mathcal{A}_i, \mathcal{B} \rangle \in \text{Trans}(\simeq_{\mathfrak{T}})$  and we conclude with the same constant function  $f$ ,  $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\text{Trans}(\simeq_{\mathfrak{T}}))$ . □

**Lemma 3.10 (Equality of non-union types is invertible)** *Let  $\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{B}$  be two non-union types.*

- (i) *If  $\mathcal{A} = a$ , then  $\mathcal{B} = a$ .*
- (ii) *If  $\mathcal{A} = \mathcal{D} @ \mathcal{A}'$ , then  $\mathcal{B} = \mathcal{D}' @ \mathcal{B}'$  with  $\mathcal{D} \simeq_{\mathfrak{T}} \mathcal{D}'$  and  $\mathcal{A}' \simeq_{\mathfrak{T}} \mathcal{B}'$ .*
- (iii) *If  $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$ , then  $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$  with  $\mathcal{A}' \simeq_{\mathfrak{T}} \mathcal{B}'$  and  $\mathcal{A}'' \simeq_{\mathfrak{T}} \mathcal{B}''$ .*

**Proof.** Immediate from the definition of subtyping. Note that there's only one applicable rule in each case. □

Along the document we often resort to the following definition and properties of the substitution operator over infinite trees:

**Definition 3.11** The **substitution** of a variable  $V$  by a tree  $\mathcal{B}$  in  $\mathcal{A}$  (notation  $\{\mathcal{B}/V\}\mathcal{A}$ ) is defined as:

$$\begin{aligned} (\{\mathcal{B}/V\}\mathcal{A})(\pi) &\triangleq \mathcal{A}(\pi) \quad \text{if } \mathcal{A}(\pi) \text{ defined and } \mathcal{A}(\pi) \neq V \\ (\{\mathcal{B}/V\}\mathcal{A})(\pi\pi') &\triangleq \mathcal{B}(\pi') \quad \text{if } \mathcal{A}(\pi) \text{ defined and } \mathcal{A}(\pi) = V \end{aligned}$$

The following lemma provides a more convenient characterisation of the substitution.

**Lemma 3.12** (i)  $\{\mathcal{B}/V\}V = \mathcal{B}$ .



- (ii)  $\{\mathcal{B}/V\} a = a$  for  $V \neq a \in \mathcal{V} \cup \mathcal{C}$ .  
 (iii)  $\{\mathcal{B}/V\} (\mathcal{A}_1 \star \mathcal{A}_2) = \{\mathcal{B}/V\} \mathcal{A}_1 \star \{\mathcal{B}/V\} \mathcal{A}_2$  for  $\star \in \{ @, \supset, \oplus \}$ .

**Proof.** The three cases are by analysis of the defined positions.

- (i) The only defined position in  $V$  is  $\epsilon$ . Then, for every  $\pi$  in  $\mathcal{B}$  we have

$$(\{\mathcal{B}/V\} V)(\pi) = (\{\mathcal{B}/V\} V)(\epsilon\pi) = B(\pi)$$

- (ii) The only defined position in  $a \neq V$  is  $\epsilon$ , thus we have  $(\{\mathcal{B}/V\} a)(\epsilon) = a(\epsilon) = a$ . Any other position is undefined.  
 (iii) Here we have  $\mathcal{A} = \mathcal{A}_1 \star \mathcal{A}_2$  with  $\star \in \{ @, \supset, \oplus \}$ . We proceed by analysing the defined positions of  $\mathcal{A}$ .

- $\pi = \epsilon$ . Then

$$(\{\mathcal{B}/V\} (\mathcal{A}_1 \star \mathcal{A}_2))(\epsilon) = (\mathcal{A}_1 \star \mathcal{A}_2)(\epsilon) = \star = (\{\mathcal{B}/V\} \mathcal{A}_1 \star \{\mathcal{B}/V\} \mathcal{A}_2)(\epsilon)$$

- $\pi = i\pi'$ . Here we have two possibilities:

- (a) either  $\mathcal{A}(\pi) \neq V$ . Then  $\mathcal{A}_i(\pi') \neq V$  and we have

$$\begin{aligned} (\{\mathcal{B}/V\} (\mathcal{A}_1 \star \mathcal{A}_2))(\pi) &= (\mathcal{A}_1 \star \mathcal{A}_2)(i\pi') && \text{by Def. 3.11} \\ &= \mathcal{A}_i(\pi') \\ &= (\{\mathcal{B}/V\} \mathcal{A}_i)(\pi') && \text{by Def. 3.11} \\ &= (\{\mathcal{B}/V\} \mathcal{A}_1 \star \{\mathcal{B}/V\} \mathcal{A}_2)(\pi) \end{aligned}$$

- (b) or  $\mathcal{A}(\pi) = V$ . Then  $\mathcal{A}_i(\pi') = V$  and by definition of substitution we have, for every position  $\pi''$  in  $\mathcal{B}$

$$\begin{aligned} (\{\mathcal{B}/V\} (\mathcal{A}_1 \star \mathcal{A}_2))(\pi\pi'') &= \mathcal{B}(\pi'') \\ &= (\{\mathcal{B}/V\} \mathcal{A}_i)(\pi'\pi'') \\ &= (\{\mathcal{B}/V\} \mathcal{A}_1 \star \{\mathcal{B}/V\} \mathcal{A}_2)(\pi\pi'') \end{aligned}$$

□

We show next that the substitution preserves the equivalent relation.

**Lemma 3.13** *Let  $\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{A}'$  and  $\mathcal{B} \simeq_{\mathfrak{T}} \mathcal{B}'$ . Then  $\{\mathcal{B}/V\} \mathcal{A} \simeq_{\mathfrak{T}} \{\mathcal{B}'/V\} \mathcal{A}'$ .*

**Proof.** Let  $\mathcal{S} = \{ \langle \{\mathcal{B}/V\} \mathcal{A}, \{\mathcal{B}'/V\} \mathcal{A}' \rangle \mid \mathcal{A} \simeq_{\mathfrak{T}} \mathcal{A}', \mathcal{B} \simeq_{\mathfrak{T}} \mathcal{B}' \}$ . We show that  $\mathcal{S} \cup \simeq_{\mathfrak{T}}$  is  $\Phi_{\simeq_{\mathfrak{T}}}$ -dense.

Let  $\langle \mathcal{C}, \mathcal{C}' \rangle \in \mathcal{S} \cup \simeq_{\mathfrak{T}}$ . If  $\langle \mathcal{C}, \mathcal{C}' \rangle \in \simeq_{\mathfrak{T}}$  the result is immediate by monotonicity of  $\Phi_{\simeq_{\mathfrak{T}}}$ , since  $\simeq_{\mathfrak{T}} = \Phi_{\simeq_{\mathfrak{T}}}(\simeq_{\mathfrak{T}}) \subseteq \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S} \cup \simeq_{\mathfrak{T}})$ . Then we only present the case where  $\langle \mathcal{C}, \mathcal{C}' \rangle \in \mathcal{S}$ ,  $\mathcal{C} = \{\mathcal{B}/V\} \mathcal{A}$  and  $\mathcal{C}' = \{\mathcal{B}'/V\} \mathcal{A}'$  with  $\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{A}'$  and  $\mathcal{B} \simeq_{\mathfrak{T}} \mathcal{B}'$ . Assume, without loss of generality

$$\begin{aligned} \mathcal{A} &= \bigoplus_{i \in 1..n} \mathcal{A}_i \quad \text{with} \quad \mathcal{A}_i \neq \oplus, i \in 1..n \\ \mathcal{A}' &= \bigoplus_{j \in 1..m} \mathcal{A}'_j \quad \text{with} \quad \mathcal{A}'_j \neq \oplus, j \in 1..m \end{aligned}$$

- (i) If  $n = m = 1$  (i.e.  $\mathcal{A}, \mathcal{A}' \neq \oplus$ ), we analyze the shape of  $\mathcal{A}$ :
- $\mathcal{A} = a$ . By Lem. 3.10,  $\mathcal{A}' = a$  and we have two possible cases. If  $a \neq V$ , by Lem. 3.12 (ii),  $\mathcal{C} = a = \mathcal{C}'$ . If not, by Lem. 3.12 (i),  $\mathcal{C} = \mathcal{B} \simeq_{\mathfrak{T}} \mathcal{B}' = \mathcal{C}'$ . Both cases are immediate by definition of  $\simeq_{\mathfrak{T}} \subseteq \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S} \cup \simeq_{\mathfrak{T}})$ .
  - $\mathcal{A} = \mathcal{D} @ \mathcal{A}_1$ . By Lem. 3.10,  $\mathcal{A}' = \mathcal{D}' @ \mathcal{A}'_1$  with  $\mathcal{D} \simeq_{\mathfrak{T}} \mathcal{D}'$  and  $\mathcal{A}_1 \simeq_{\mathfrak{T}} \mathcal{A}'_1$ . Then, by definition of  $\mathcal{S}$ , we have  $\langle \{\mathcal{B}/V\} \mathcal{D}, \{\mathcal{B}'/V\} \mathcal{D}' \rangle$  and  $\langle \{\mathcal{B}/V\} \mathcal{A}_1, \{\mathcal{B}'/V\} \mathcal{A}'_1 \rangle \in \mathcal{S} \cup \simeq_{\mathfrak{T}}$ . Finally we conclude  $\langle \mathcal{C}, \mathcal{C}' \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S} \cup \simeq_{\mathfrak{T}})$  since, by Lem. 3.12 (iii),

$$\begin{aligned} \mathcal{C} &= \{\mathcal{B}/V\} (\mathcal{D} @ \mathcal{A}_1) = \{\mathcal{B}/V\} \mathcal{D} @ \{\mathcal{B}/V\} \mathcal{A}_1 \\ \mathcal{C}' &= \{\mathcal{B}'/V\} (\mathcal{D}' @ \mathcal{A}'_1) = \{\mathcal{B}'/V\} \mathcal{D}' @ \{\mathcal{B}'/V\} \mathcal{A}'_1 \end{aligned}$$

- $\mathcal{A} = \mathcal{A}_1 \supset \mathcal{A}_2$ . As before, by Lem. 3.10, we get  $\mathcal{A} = \mathcal{A}'_1 \supset \mathcal{A}'_2$  with  $\mathcal{A}_1 \simeq_{\mathfrak{T}} \mathcal{A}'_1$  and  $\mathcal{A}_2 \simeq_{\mathfrak{T}} \mathcal{A}'_2$ . By definition  $\mathcal{S}$  we have  $\langle \{\mathcal{B}/V\} \mathcal{A}_1, \{\mathcal{B}'/V\} \mathcal{A}'_1 \rangle$  and  $\langle \{\mathcal{B}/V\} \mathcal{A}_2, \{\mathcal{B}'/V\} \mathcal{A}'_2 \rangle \in \mathcal{S} \cup \simeq_{\mathfrak{T}}$ . Thus, we conclude by Lem. 3.12 (iii),  $\langle \mathcal{C}, \mathcal{C}' \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S} \cup \simeq_{\mathfrak{T}})$ .

- (ii) If  $n + m > 2$ , by (E-UNION-T) we have

$$\begin{aligned} \exists f : 1..n \rightarrow 1..m \quad \text{s.t.} \quad \mathcal{A}_i \simeq_{\mathfrak{T}} \mathcal{A}'_{f(i)} \quad \text{for every } i \in 1..n \\ \exists g : 1..m \rightarrow 1..n \quad \text{s.t.} \quad \mathcal{A}_{g(j)} \simeq_{\mathfrak{T}} \mathcal{A}'_j \quad \text{for every } j \in 1..m \end{aligned}$$

Then,  $\langle \{\mathcal{B}/V\} \mathcal{A}_i, \{\mathcal{B}'/V\} \mathcal{A}'_{f(i)} \rangle$  and  $\langle \{\mathcal{B}/V\} \mathcal{A}_{g(j)}, \{\mathcal{B}'/V\} \mathcal{A}'_j \rangle \in \mathcal{S} \cup \simeq_{\mathfrak{T}}$  for every  $i \in 1..n, j \in 1..m$ . Once again we conclude by definition of  $\Phi_{\simeq_{\mathfrak{T}}}$  and Lem. 3.12 (iii),  $\langle \mathcal{C}, \mathcal{C}' \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S} \cup \simeq_{\mathfrak{T}})$ . □

### 3.1.3 Subtyping of trees

In a similar way we have a coinductive characterization of subtyping over trees.

**Definition 3.14** Infinite type subtyping, written  $\preceq_{\mathfrak{T}}$ , is defined by the coinductive interpretation of the schemes in Fig. 4.

The most interesting rule in Fig. 4 is (S-UNION-T). Here, for a maximal union type of the form  $\bigoplus_{i \in 1..n} \mathcal{A}_i$  to be a subtype of a maximal union type  $\bigoplus_{j \in 1..m} \mathcal{B}_j$ , one of the two must have at least one occurrence of the union type construct ( $n + m > 2$ ) and there must be a function  $f : 1..n \rightarrow 1..m$  such that  $\mathcal{A}_i \preceq_{\mathfrak{T}} \mathcal{B}_{f(i)}$  for each  $i \in 1..n$ .

**Remark 3.15** The rules are derived from those of Fig. 2. More precisely, rules (S-UNION-R1), (S-UNION-R2) and (S-UNION-L) of Fig. 2 and the observation that (S-UNION-R1) and (S-UNION-R2) can always be permuted past (S-UNION-L).

As above, the formal definition of the subtyping relation is given by the associated

$$\begin{array}{c}
 \frac{}{a \preceq_{\mathfrak{T}} a} \text{ (S-REFL-T)} \\
 \\
 \frac{\mathcal{A}' \preceq_{\mathfrak{T}} \mathcal{A} \quad \mathcal{B} \preceq_{\mathfrak{T}} \mathcal{B}'}{\mathcal{A} \supset \mathcal{B} \preceq_{\mathfrak{T}} \mathcal{A}' \supset \mathcal{B}'} \text{ (S-FUNC-T)} \quad \frac{\mathcal{D} \preceq_{\mathfrak{T}} \mathcal{D}' \quad \mathcal{A} \preceq_{\mathfrak{T}} \mathcal{A}'}{\mathcal{D} @ \mathcal{A} \preceq_{\mathfrak{T}} \mathcal{D}' @ \mathcal{A}'} \text{ (S-COMP-T)} \\
 \\
 \frac{\mathcal{A}_i \preceq_{\mathfrak{T}} \mathcal{B}_{f(i)} \quad f : 1..n \rightarrow 1..m \quad \mathcal{A}_i, \mathcal{B}_j \neq \oplus \quad n + m > 2}{\bigoplus_{i \in 1..n} \mathcal{A}_i \preceq_{\mathfrak{T}} \bigoplus_{j \in 1..m} \mathcal{B}_j} \text{ (S-UNION-T)}
 \end{array}$$

Fig. 4. Subtyping relation for infinite types

function  $\Phi_{\preceq_{\mathfrak{T}}} : \wp(\mathfrak{T} \times \mathfrak{T}) \rightarrow \wp(\mathfrak{T} \times \mathfrak{T})$  defined next:

$$\begin{aligned}
 \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{S}) = & \{ \langle a, a \rangle \mid a \in \mathcal{V} \cup \mathcal{C} \} \\
 & \cup \{ \langle \mathcal{D} @ \mathcal{A}, \mathcal{D}' @ \mathcal{A}' \rangle \mid \langle \mathcal{D}, \mathcal{D}' \rangle, \langle \mathcal{A}, \mathcal{A}' \rangle \in \mathcal{S} \} \\
 & \cup \{ \langle \mathcal{A} \supset \mathcal{B}, \mathcal{A}' \supset \mathcal{B}' \rangle \mid \langle \mathcal{A}', \mathcal{A} \rangle, \langle \mathcal{B}, \mathcal{B}' \rangle \in \mathcal{S} \} \\
 & \cup \{ \langle \bigoplus_{i \in 1..n} \mathcal{A}_i, \bigoplus_{j \in 1..m} \mathcal{B}_j \rangle \mid \mathcal{A}_i, \mathcal{B}_j \neq \oplus, n + m > 2 \\
 & \quad \exists f : 1..n \rightarrow 1..m \text{ s.t. } \langle \mathcal{A}_i, \mathcal{B}_{f(i)} \rangle \in \mathcal{S} \}
 \end{aligned}$$

Then  $\preceq_{\mathfrak{T}} = \nu \Phi_{\preceq_{\mathfrak{T}}}$ . We now address some properties of subtyping.

**Lemma 3.16 (Subtyping is a preorder)**  $\preceq_{\mathfrak{T}}$  is a preorder (i.e. reflexive and transitive).

**Proof.** This proof is similar to the one presented before for  $\simeq_{\mathfrak{T}}$ .  $\square$

The following notion of invertibility (Lem. 3.17) is the main result of the present Section and an essential property to prove Subject Reduction (Prop. 4.1) and Progress (Prop. 4.3) for the type system proposed in Sec. 3.

**Lemma 3.17 (Subtyping of non-union types is invertible)** Let  $\mathcal{A}, \mathcal{B} \in \mathfrak{T}$  be non-union types. Suppose  $\mathcal{A} \preceq_{\mathfrak{T}} \mathcal{B}$ .

- (i) If  $\mathcal{A} = a$ , then  $\mathcal{B} = a$ .
- (ii) If  $\mathcal{A} = \mathcal{D} @ \mathcal{A}'$ , then  $\mathcal{B} = \mathcal{D}' @ \mathcal{B}'$  with  $\mathcal{D} \preceq_{\mathfrak{T}} \mathcal{D}'$  and  $\mathcal{A}' \preceq_{\mathfrak{T}} \mathcal{B}'$ .
- (iii) If  $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$ , then  $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$  with  $\mathcal{B}' \preceq_{\mathfrak{T}} \mathcal{A}'$  and  $\mathcal{A}'' \preceq_{\mathfrak{T}} \mathcal{B}''$ .

**Remark 3.18** In each of the three items of Lem. 3.17 the roles of  $\mathcal{A}$  and  $\mathcal{B}$  can be reversed.

**Lemma 3.19**  $\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{B} \implies \mathcal{A} \preceq_{\mathfrak{T}} \mathcal{B}$ .

**Proof.** We show that  $\simeq_{\mathfrak{T}} = \Phi_{\simeq_{\mathfrak{T}}}(\simeq_{\mathfrak{T}})$  is  $\Phi_{\preceq_{\mathfrak{T}}}$ -dense. Let  $\langle \mathcal{A}, \mathcal{B} \rangle \in \simeq_{\mathfrak{T}}$ . By Rem. 3.7 we can consider maximal union types

$$\begin{aligned}
 \mathcal{A} &= \bigoplus_{i \in 1..n} \mathcal{A}_i \quad \text{with} \quad \mathcal{A}_i \neq \oplus, i \in 1..n \\
 \mathcal{B} &= \bigoplus_{j \in 1..m} \mathcal{B}_j \quad \text{with} \quad \mathcal{B}_j \neq \oplus, j \in 1..m
 \end{aligned}$$

and we have two separate cases to analyze:

- (i) If  $n = m = 1$ , then both  $\mathcal{A}$  and  $\mathcal{B}$  are non-union types. Now we proceed by analyzing the shape of  $\mathcal{A}$ :
  - $\mathcal{A} = a$ . Then, by definition of  $\Phi_{\simeq_{\mathfrak{T}}}$ ,  $\mathcal{B} = a$  and the result is immediate since  $\langle a, a \rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\simeq_{\mathfrak{T}})$  for every  $a \in \mathcal{V} \cup \mathcal{C}$ .
  - $\mathcal{A} = \mathcal{D} @ \mathcal{A}'$ . Again, by definition of  $\Phi_{\simeq_{\mathfrak{T}}}$ , we have  $\mathcal{B} = \mathcal{D}' @ \mathcal{B}'$  with  $\langle \mathcal{D}, \mathcal{D}' \rangle, \langle \mathcal{A}', \mathcal{B}' \rangle \in \simeq_{\mathfrak{T}}$ . Then we conclude by definition of  $\Phi_{\preceq_{\mathfrak{T}}}$ ,  $\langle \mathcal{D} @ \mathcal{A}', \mathcal{D}' @ \mathcal{B}' \rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\simeq_{\mathfrak{T}})$ .
  - $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$ . Similarly,  $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$  with  $\langle \mathcal{A}', \mathcal{B}' \rangle, \langle \mathcal{A}'', \mathcal{B}'' \rangle \in \simeq_{\mathfrak{T}}$ . By symmetry  $\langle \mathcal{B}', \mathcal{A}' \rangle \in \simeq_{\mathfrak{T}}$  and we conclude  $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\simeq_{\mathfrak{T}})$ .
- (ii) If not (*i.e.*  $n + m > 2$ ), rule (E-UNION-T) applies. Then

$$\begin{aligned} \exists f : 1..n \rightarrow 1..m \quad \text{s.t.} \quad \langle \mathcal{A}_i, \mathcal{B}_{f(i)} \rangle \in \simeq_{\mathfrak{T}} \quad \text{for every } i \in 1..n \\ \exists g : 1..m \rightarrow 1..n \quad \text{s.t.} \quad \langle \mathcal{A}_{g(j)}, \mathcal{B}_j \rangle \in \simeq_{\mathfrak{T}} \quad \text{for every } j \in 1..m \end{aligned}$$

Thus, we conclude with the same function  $f$ ,  $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\simeq_{\mathfrak{T}})$ . □

To prove the correspondence of the coinductive formulation with the inductive approach, it is convenient to work with finite trees (types). Thus, we introduce a characterisation of the equivalence and subtyping relations in terms of finite truncations of infinite trees.

We denote with  $\#_{\oplus}(\mathcal{A})$  the maximal number of adjacent union type nodes, starting from the root of  $\mathcal{A}$ :

$$\#_{\oplus}(\mathcal{A}) \triangleq \begin{cases} 0 & \text{if } \mathcal{A} \neq \oplus \\ 1 + \#_{\oplus}(\mathcal{A}_1) + \#_{\oplus}(\mathcal{A}_2) & \text{if } \mathcal{A} = \mathcal{A}_1 \oplus \mathcal{A}_2 \end{cases}$$

Recall that, by definition of  $\mathfrak{T}$ , a type cannot consist of infinitely many consecutive occurrences of  $\oplus$ . Thus, the previous inductive definition is well-founded, as well as the following:

**Definition 3.20** The **truncation** of a tree  $\mathcal{A}$  at depth  $k \in \mathbb{N}$  (notation  $\mathcal{A}|_k$ ) is defined inductively<sup>7</sup> as follows:

$$\begin{aligned} \mathcal{A}|_0 &\triangleq \bullet \\ a|_{k+1} &\triangleq a && \text{for } a \in \mathcal{V} \cup \mathcal{C} \\ (\mathcal{A}_1 \star \mathcal{A}_2)|_{k+1} &\triangleq \mathcal{A}_1|_k \star \mathcal{A}_2|_k && \text{for } \star \in \{ @, \supset \} \\ (\mathcal{A}_1 \oplus \mathcal{A}_2)|_{k+1} &\triangleq \mathcal{A}_1|_{k+1} \oplus \mathcal{A}_2|_{k+1} \end{aligned}$$

where  $\bullet \in \mathcal{C}$  is a distinguished type constant used to identify the nodes where the tree was truncated.

<sup>7</sup> Using the lexicographical extension of the standard order to  $\langle k, \#_{\oplus}(\mathcal{A}) \rangle$ .

**Remark 3.21** Given a maximal union type  $\bigoplus_{i \in 1..n} \mathcal{A}_i$ , immediately from the definition we have  $(\bigoplus_{i \in 1..n} \mathcal{A}_i)|_{k+1} = \bigoplus_{i \in 1..n} (\mathcal{A}_i|_{k+1})$ .

**Lemma 3.22**  $\forall k \in \mathbb{N}. \mathcal{A}|_k \simeq_{\mathfrak{T}} \mathcal{B}|_k \text{ iff } \mathcal{A} \simeq_{\mathfrak{T}} \mathcal{B}$ .

**Proof.**  $\Rightarrow$ ) We show that  $\mathcal{S} \triangleq \{\langle \mathcal{A}, \mathcal{B} \rangle \mid \forall k \in \mathbb{N}. \mathcal{A}|_k \simeq_{\mathfrak{T}} \mathcal{B}|_k\}$  is  $\Phi_{\simeq_{\mathfrak{T}}}$ -dense. Let  $\langle \mathcal{A}, \mathcal{B} \rangle \in \mathcal{S}$ . Then, for every  $k \in \mathbb{N}$  we have  $\mathcal{A}|_k \simeq_{\mathfrak{T}} \mathcal{B}|_k$ . Consider maximal union types

$$\begin{aligned} \mathcal{A} &= \bigoplus_{i \in 1..n} \mathcal{A}_i \quad \text{with} \quad \mathcal{A}_i \neq \oplus, i \in 1..n \\ \mathcal{B} &= \bigoplus_{j \in 1..m} \mathcal{B}_j \quad \text{with} \quad \mathcal{B}_j \neq \oplus, j \in 1..m \end{aligned}$$

(i) If  $n = m = 1$  (i.e.  $\mathcal{A}, \mathcal{B} \neq \oplus$ ), we proceed by analyzing the shape of  $\mathcal{A}$ :

- $\mathcal{A} = a$ . Then,  $\mathcal{A}|_k = a$  for every  $k > 0$  and, by Lem. 3.10,  $\mathcal{B}|_k = a$ . Hence,  $\mathcal{B} = a$  and we conclude directly from the definition of  $\Phi_{\simeq_{\mathfrak{T}}}$ ,  $\langle a, a \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S})$ .
- $\mathcal{A} = \mathcal{D} @ \mathcal{A}'$ . Similarly, we have  $\mathcal{A}|_k = \mathcal{D}|_{k-1} @ \mathcal{A}'|_{k-1}$  for every  $k > 0$ . By Lem. 3.10 once again, we get  $\mathcal{B}|_k = \mathcal{D}'_k @ \mathcal{B}'_k$  with  $\mathcal{D}|_{k-1} \simeq_{\mathfrak{T}} \mathcal{D}'_k$  and  $\mathcal{A}'|_{k-1} \simeq_{\mathfrak{T}} \mathcal{B}'_k$ . Note that for every  $k$  we have different subtrees  $\mathcal{D}'_k$  and  $\mathcal{B}'_k$  but, since Lem. 3.10 refers to tree equality (not equivalence) when determining the shape of  $\mathcal{B}$ , it is immediate to see from the definition of the truncation that  $\mathcal{B} = \mathcal{D}' @ \mathcal{B}'$  with  $\mathcal{D}'_k = \mathcal{D}'|_{k-1}$  and  $\mathcal{B}'_k = \mathcal{B}'|_{k-1}$  for every  $k > 0$ . Hence,  $\mathcal{D}|_{k-1} \simeq_{\mathfrak{T}} \mathcal{D}'|_{k-1}$  and  $\mathcal{A}'|_{k-1} \simeq_{\mathfrak{T}} \mathcal{B}'|_{k-1}$  for every  $k > 0$ . Then, by definition of  $\mathcal{S}$ ,  $\langle \mathcal{D}, \mathcal{D}' \rangle, \langle \mathcal{A}', \mathcal{B}' \rangle \in \mathcal{S}$  and we conclude  $\langle \mathcal{D} @ \mathcal{A}', \mathcal{D}' @ \mathcal{B}' \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S})$ .
- $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$ . Analysis for this case is similar to the previous one. From  $\mathcal{A}|_k = \mathcal{A}'|_{k-1} \supset \mathcal{A}''|_{k-1}$  we get  $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$  with  $\mathcal{A}'|_{k-1} \simeq_{\mathfrak{T}} \mathcal{B}'|_{k-1}$  and  $\mathcal{A}''|_{k-1} \simeq_{\mathfrak{T}} \mathcal{B}''|_{k-1}$  for every  $k > 0$ . Then we have  $\langle \mathcal{A}', \mathcal{B}' \rangle, \langle \mathcal{A}'', \mathcal{B}'' \rangle \in \mathcal{S}$  and conclude  $\langle \mathcal{A}' \supset \mathcal{A}'', \mathcal{B}' \supset \mathcal{B}'' \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S})$ .

(ii) If  $n + m > 2$  we have  $\mathcal{A}|_k = \bigoplus_{i \in 1..n} (\mathcal{A}_i|_k)$  and  $\mathcal{B}|_k = \bigoplus_{j \in 1..m} (\mathcal{B}_j|_k)$  for every  $k > 0$ . From  $\mathcal{A}|_k \simeq_{\mathfrak{T}} \mathcal{B}|_k$ , by (E-UNION-T), we get

$$\begin{aligned} \exists f : 1..n \rightarrow 1..m \quad \text{s.t.} \quad \mathcal{A}_i|_k \simeq_{\mathfrak{T}} \mathcal{B}_{f(i)}|_k \quad \text{for every} \quad i \in 1..n \\ \exists g : 1..m \rightarrow 1..n \quad \text{s.t.} \quad \mathcal{A}_{g(j)}|_k \simeq_{\mathfrak{T}} \mathcal{B}_j|_k \quad \text{for every} \quad j \in 1..m \end{aligned}$$

Since  $\mathcal{C}|_0 = \bullet$  for every  $\mathcal{C} \in \mathfrak{T}$ , we have  $\mathcal{A}_i|_0 \simeq_{\mathfrak{T}} \mathcal{B}_{f(i)}|_0$  and  $\mathcal{A}_{g(j)}|_0 \simeq_{\mathfrak{T}} \mathcal{B}_j|_0$  by reflexivity. Thus,  $\mathcal{A}_i|_k \simeq_{\mathfrak{T}} \mathcal{B}_{f(i)}|_k$  and  $\mathcal{A}_{g(j)}|_k \simeq_{\mathfrak{T}} \mathcal{B}_j|_k$  for every  $k \in \mathbb{N}$ . Then, by definition of  $\mathcal{S}$ ,  $\langle \mathcal{A}_i, \mathcal{B}_{f(i)} \rangle, \langle \mathcal{A}_{g(j)}, \mathcal{B}_j \rangle \in \mathcal{S}$  for every  $i \in 1..n, j \in 1..m$ . Finally, we conclude  $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S})$ .

$\Leftarrow$ ) For this part of the proof we show that the converse relation  $\mathcal{S} \triangleq \{\langle \mathcal{A}|_k, \mathcal{B}|_k \rangle \mid \mathcal{A} \simeq_{\mathfrak{T}} \mathcal{B}, k \in \mathbb{N}\}$  is  $\Phi_{\simeq_{\mathfrak{T}}}$ -dense. Let  $\langle \mathcal{A}|_k, \mathcal{B}|_k \rangle \in \mathcal{S}$ . If  $k = 0$ , by definition of the truncation,  $\mathcal{A}|_k = \bullet = \mathcal{B}|_k$  and trivially  $\langle \bullet, \bullet \rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S})$ . We analyze next the cases where  $k > 0$  given that, by definition of  $\mathcal{S}$ ,  $\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{B}$ . Once again we consider maximal union types

$$\begin{aligned} \mathcal{A} &= \bigoplus_{i \in 1..n} \mathcal{A}_i \quad \text{with} \quad \mathcal{A}_i \neq \oplus, i \in 1..n \\ \mathcal{B} &= \bigoplus_{j \in 1..m} \mathcal{B}_j \quad \text{with} \quad \mathcal{B}_j \neq \oplus, j \in 1..m \end{aligned}$$

and analyze separately the cases where both  $\mathcal{A}$  and  $\mathcal{B}$  are non-union types.

- (i) If  $n = m = 1$  we look at the shape of  $\mathcal{A}$ :
- $\mathcal{A} = a$ . By Lem. 3.10,  $\mathcal{B} = a$  and  $a|_k = a$  for every  $k > 0$ . Then we conclude by definition of  $\Phi_{\simeq_{\mathcal{T}}}$ ,  $\langle a, a \rangle \in \Phi_{\simeq_{\mathcal{T}}}(\mathcal{S})$ .
  - $\mathcal{A} = \mathcal{D} @ \mathcal{A}'$ . By Lem. 3.10,  $\mathcal{B} = \mathcal{D}' @ \mathcal{B}'$  with  $\mathcal{D} \simeq_{\mathcal{T}} \mathcal{D}'$  and  $\mathcal{A}' \simeq_{\mathcal{T}} \mathcal{B}'$ . Then, by definition of  $\mathcal{S}$ ,  $\langle \mathcal{D}|_{k-1}, \mathcal{D}'|_{k-1} \rangle, \langle \mathcal{A}'|_{k-1}, \mathcal{B}'|_{k-1} \rangle \in \mathcal{S}$  and we conclude  $\langle \mathcal{A}|_k, \mathcal{B}|_k \rangle = \langle \mathcal{D}|_{k-1} @ \mathcal{A}'|_{k-1}, \mathcal{D}'|_{k-1} @ \mathcal{B}'|_{k-1} \rangle \in \Phi_{\simeq_{\mathcal{T}}}(\mathcal{S})$ .
  - $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$ . Similarly to the previous case, we have  $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$  with  $\mathcal{A}' \simeq_{\mathcal{T}} \mathcal{B}'$  and  $\mathcal{A}'' \simeq_{\mathcal{T}} \mathcal{B}''$ . Then  $\langle \mathcal{A}'|_{k-1}, \mathcal{B}'|_{k-1} \rangle, \langle \mathcal{A}''|_{k-1}, \mathcal{B}''|_{k-1} \rangle \in \mathcal{S}$  and  $\langle \mathcal{A}|_k, \mathcal{B}|_k \rangle = \langle \mathcal{A}'|_{k-1} \supset \mathcal{A}''|_{k-1}, \mathcal{B}'|_{k-1} \supset \mathcal{B}''|_{k-1} \rangle \in \Phi_{\simeq_{\mathcal{T}}}(\mathcal{S})$ .
- (ii) If  $n + m > 2$ , by (E-UNION-T) we have

$$\begin{aligned} \exists f : 1..n \rightarrow 1..m \quad \text{s.t.} \quad \mathcal{A}_i \simeq_{\mathcal{T}} \mathcal{B}_{f(i)} \quad \text{for every } i \in 1..n \\ \exists g : 1..m \rightarrow 1..n \quad \text{s.t.} \quad \mathcal{A}_{g(j)} \simeq_{\mathcal{T}} \mathcal{B}_j \quad \text{for every } j \in 1..m \end{aligned}$$

Then, by definition of  $\mathcal{S}$ ,  $\langle \mathcal{A}_i|_k, \mathcal{B}_{f(i)}|_k \rangle, \langle \mathcal{A}_{g(j)}|_k, \mathcal{B}_j|_k \rangle \in \mathcal{S}$  for every  $k > 0$ . Thus, we conclude by resorting to Rem. 3.21,  $\langle \mathcal{A}|_k, \mathcal{B}|_k \rangle \in \Phi_{\simeq_{\mathcal{T}}}(\mathcal{S})$ .  $\square$

**Lemma 3.23**  $\forall k \in \mathbb{N}. \mathcal{A}|_k \preceq_{\mathcal{T}} \mathcal{B}|_k \text{ iff } \mathcal{A} \preceq_{\mathcal{T}} \mathcal{B}$ .

**Proof.**  $\Rightarrow$ ) Similarly to the previous lemma, we prove this part by showing that  $\mathcal{S} \triangleq \{ \langle \mathcal{A}, \mathcal{B} \rangle \mid \forall k \in \mathbb{N}. \mathcal{A}|_k \preceq_{\mathcal{T}} \mathcal{B}|_k \}$  is  $\Phi_{\preceq_{\mathcal{T}}}$ -dense. By hypothesis we have  $\mathcal{A}|_k \preceq_{\mathcal{T}} \mathcal{B}|_k$  for every  $k \in \mathbb{N}$ . As before we consider maximal union types and analyze separately the case for non-union types

$$\begin{aligned} \mathcal{A} &= \bigoplus_{i \in 1..n} \mathcal{A}_i \quad \text{with} \quad \mathcal{A}_i \neq \oplus, i \in 1..n \\ \mathcal{B} &= \bigoplus_{j \in 1..m} \mathcal{B}_j \quad \text{with} \quad \mathcal{B}_j \neq \oplus, j \in 1..m \end{aligned}$$

- (i) If  $n = m = 1$  (i.e.  $\mathcal{A}, \mathcal{B} \neq \oplus$ ), we proceed by analyzing the shape of  $\mathcal{A}$ :
- $\mathcal{A} = a$ . Then,  $\mathcal{A}|_k = a$  for every  $k > 0$  and, by Lem. 3.17,  $\mathcal{B}|_k = a$ . Hence,  $\mathcal{B} = a$  and we conclude directly from the definition of  $\Phi_{\preceq_{\mathcal{T}}}$ ,  $\langle a, a \rangle \in \Phi_{\preceq_{\mathcal{T}}}(\mathcal{S})$ .
  - $\mathcal{A} = \mathcal{D} @ \mathcal{A}'$ . Similarly, we have  $\mathcal{A}|_k = \mathcal{D}|_{k-1} @ \mathcal{A}'|_{k-1}$  for every  $k > 0$ . By Lem. 3.17 once again, we get  $\mathcal{B}|_k = \mathcal{D}'_k @ \mathcal{B}'_k$  with  $\mathcal{D}|_{k-1} \preceq_{\mathcal{T}} \mathcal{D}'_k$  and  $\mathcal{A}'|_{k-1} \preceq_{\mathcal{T}} \mathcal{B}'_k$ . As in the previous lemma, in this case we have different subtrees  $\mathcal{D}'_k$  and  $\mathcal{B}'_k$  for every  $k$  but, by resorting to tree equality on Lem. 3.17 and the definition of the truncation, we can assure that  $\mathcal{B} = \mathcal{D}' @ \mathcal{B}'$  with  $\mathcal{D}'_k = \mathcal{D}'|_{k-1}$  and  $\mathcal{B}'_k = \mathcal{B}'|_{k-1}$  for every  $k > 0$ . Hence,  $\mathcal{D}|_{k-1} \preceq_{\mathcal{T}} \mathcal{D}'|_{k-1}$  and  $\mathcal{A}'|_{k-1} \preceq_{\mathcal{T}} \mathcal{B}'|_{k-1}$  for every  $k > 0$ . Then, by definition of  $\mathcal{S}$ ,  $\langle \mathcal{D}, \mathcal{D}' \rangle, \langle \mathcal{A}', \mathcal{B}' \rangle \in \mathcal{S}$  and we conclude  $\langle \mathcal{D} @ \mathcal{A}', \mathcal{D}' @ \mathcal{B}' \rangle \in \Phi_{\preceq_{\mathcal{T}}}(\mathcal{S})$ .
  - $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$ . Analysis for this case is similar to the previous one. From  $\mathcal{A}|_k = \mathcal{A}'|_{k-1} \supset \mathcal{A}''|_{k-1}$  we get  $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$  with  $\mathcal{B}'|_{k-1} \preceq_{\mathcal{T}} \mathcal{A}'|_{k-1}$  and  $\mathcal{A}''|_{k-1} \preceq_{\mathcal{T}} \mathcal{B}''|_{k-1}$  for every  $k > 0$ . Note that, by Lem. 3.17, subtyping order on the domains is inverted. Then we have  $\langle \mathcal{B}', \mathcal{A}' \rangle, \langle \mathcal{A}'', \mathcal{B}'' \rangle \in \mathcal{S}$  and conclude  $\langle \mathcal{A}' \supset \mathcal{A}'', \mathcal{B}' \supset \mathcal{B}'' \rangle \in \Phi_{\preceq_{\mathcal{T}}}(\mathcal{S})$ .

- (ii) If  $n + m > 2$  we have  $\mathcal{A}|_k = \bigoplus_{i \in 1..n} (\mathcal{A}_i|_k)$  and  $\mathcal{B}|_k = \bigoplus_{j \in 1..m} (\mathcal{B}_j|_k)$  for every  $k > 0$ . From  $\mathcal{A}|_k \preceq_{\mathfrak{T}} \mathcal{B}|_k$ , by (S-UNION-T), we get

$$\begin{aligned} \exists f : 1..n \rightarrow 1..m \quad \text{s.t.} \quad \mathcal{A}_i|_k \preceq_{\mathfrak{T}} \mathcal{B}_{f(i)}|_k \quad \text{for every } i \in 1..n \\ \exists g : 1..m \rightarrow 1..n \quad \text{s.t.} \quad \mathcal{A}_{g(j)}|_k \preceq_{\mathfrak{T}} \mathcal{B}_j|_k \quad \text{for every } j \in 1..m \end{aligned}$$

Since  $\mathcal{C}|_0 = \bullet$  for every  $\mathcal{C} \in \mathfrak{T}$ , we also have  $\mathcal{A}_i|_0 \preceq_{\mathfrak{T}} \mathcal{B}_{f(i)}|_0$  and  $\mathcal{A}_{g(j)}|_0 \preceq_{\mathfrak{T}} \mathcal{B}_j|_0$  by reflexivity. Thus,  $\mathcal{A}_i|_k \preceq_{\mathfrak{T}} \mathcal{B}_{f(i)}|_k$  and  $\mathcal{A}_{g(j)}|_k \preceq_{\mathfrak{T}} \mathcal{B}_j|_k$  for every  $k \in \mathbb{N}$ . Then, by definition of  $\mathcal{S}$ ,  $\langle \mathcal{A}_i, \mathcal{B}_{f(i)} \rangle, \langle \mathcal{A}_{g(j)}, \mathcal{B}_j \rangle \in \mathcal{S}$  for every  $i \in 1..n, j \in 1..m$ . Finally, we conclude  $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{S})$ .

$\Leftarrow$ ) As before, we define  $\mathcal{S} \triangleq \{ \langle \mathcal{A}|_k, \mathcal{B}|_k \rangle \mid \mathcal{A} \preceq_{\mathfrak{T}} \mathcal{B}, k \in \mathbb{N} \}$  and show that is  $\Phi_{\preceq_{\mathfrak{T}}}$ -dense to prove this part of the lemma. Again, if  $k = 0$  the result is immediate, so let's focus on the case where  $k > 0$ .

Let  $\mathcal{A} \preceq_{\mathfrak{T}} \mathcal{B}$ . We assume, without loss of generality,  $\mathcal{A} = \bigoplus_{i \in 1..n} \mathcal{A}_i$  and  $\mathcal{B} = \bigoplus_{j \in 1..m} \mathcal{B}_j$  are maximal union types.

If  $n + m > 2$  it is the case of (S-UNION-T) and we have  $\exists f : 1..n \rightarrow 1..m$  such that  $\mathcal{A}_i \preceq_{\mathfrak{T}} \mathcal{B}_{f(i)}$  for every  $i \in 1..n$ . Then, by definition we have  $\langle \mathcal{A}_i|_k, \mathcal{B}_{f(i)}|_k \rangle \in \mathcal{S}$  and conclude  $\langle \mathcal{A}|_k, \mathcal{B}|_k \rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{S})$ .

On the other hand, if  $n = 1 = m$  we analyze the form of  $\mathcal{A}$ :

- (i)  $\mathcal{A} = a$ . By Lem. 3.17 we have  $\mathcal{B} = a$  and the result is immediate.
- (ii)  $\mathcal{A} = \mathcal{D} @ \mathcal{A}'$ . By Lem. 3.17,  $\mathcal{B} = \mathcal{D}' @ \mathcal{B}'$  with  $\mathcal{D} \preceq_{\mathfrak{T}} \mathcal{D}'$  and  $\mathcal{A}' \preceq_{\mathfrak{T}} \mathcal{B}'$ . Then we have  $\langle \mathcal{D}|_{k-1}, \mathcal{D}'|_{k-1} \rangle, \langle \mathcal{A}'|_{k-1}, \mathcal{B}'|_{k-1} \rangle \in \mathcal{S}$  for every  $k > 0$ , and conclude by definition of  $\Phi_{\preceq_{\mathfrak{T}}}$ ,  $\langle \mathcal{A}|_k, \mathcal{B}|_k \rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{S})$ .
- (iii)  $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$ . Similarly to the previous case we have  $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$  with  $\mathcal{B}' \preceq_{\mathfrak{T}} \mathcal{A}'$  and  $\mathcal{A}'' \preceq_{\mathfrak{T}} \mathcal{B}''$ . Then we conclude by definition of  $\mathcal{S}$  and  $\Phi_{\preceq_{\mathfrak{T}}}$  that  $\langle \mathcal{A}|_k, \mathcal{B}|_k \rangle = \langle \mathcal{A}'|_{k-1} \supset \mathcal{A}''|_{k-1}, \mathcal{B}'|_{k-1} \supset \mathcal{B}''|_{k-1} \rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{S})$ .

□

### 3.1.4 Correspondence between $\mu$ -types and infinite types

Contractive  $\mu$ -types characterize [1, 7, 16, 25] a proper subset of  $\mathfrak{T}$  known as the **regular trees** (trees whose set of distinct subtrees is finite) and denoted  $\mathfrak{T}^{reg}$ . Given a contractive  $\mu$ -type  $A$ ,  $\llbracket A \rrbracket^{\mathfrak{T}}$  is the regular tree obtained by completely unfolding all occurrences of  $\mu V.B$  in  $A$ . Def. 3.24 below extends that of [25] to union and data types. It is well-founded, relying on the lexicographical extension of the standard order to  $\langle |\pi|, \#_{\mu}(A) \rangle$ , where  $\#_{\mu}(A)$  is the number of occurrences of the  $\mu$  type constructor at the head position of  $A$ .

**Definition 3.24** The function  $\llbracket \bullet \rrbracket^{\mathfrak{T}} : \mathcal{T} \rightarrow \mathfrak{T}^{reg}$ , mapping  $\mu$ -types to types, is de-



defined inductively as follows:

$$\begin{aligned}
 \llbracket a \rrbracket^{\mathfrak{T}}(\epsilon) &\triangleq a \\
 \llbracket A_1 \star A_2 \rrbracket^{\mathfrak{T}}(\epsilon) &\triangleq \star && \text{for } \star \in \{\textcircled{A}, \supset, \oplus\} \\
 \llbracket A_1 \star A_2 \rrbracket^{\mathfrak{T}}(i\pi) &\triangleq \llbracket A_i \rrbracket^{\mathfrak{T}}(\pi) && \text{for } \star \in \{\textcircled{A}, \supset, \oplus\} \\
 \llbracket \mu V.A \rrbracket^{\mathfrak{T}}(\pi) &\triangleq \llbracket \{\mu V.A/V\} A \rrbracket^{\mathfrak{T}}(\pi)
 \end{aligned}$$

Commutation of  $\llbracket \bullet \rrbracket^{\mathfrak{T}}$  with substitutions is as expected.

**Lemma 3.25**  $\llbracket \{B/V\} A \rrbracket^{\mathfrak{T}} = \{\llbracket B \rrbracket^{\mathfrak{T}}/V\} \llbracket A \rrbracket^{\mathfrak{T}}.$

**Proof.** We actually prove the equivalent result

$$\forall k \in \mathbb{N}. \llbracket \{B/V\} A \rrbracket^{\mathfrak{T}}|_k = (\{\llbracket B \rrbracket^{\mathfrak{T}}/V\} \llbracket A \rrbracket^{\mathfrak{T}})|_k$$

and conclude by reflexivity of  $\simeq_{\mathfrak{T}}$  and Lem. 3.22.

The proof is by induction on the lexicographical extension of the standard order to  $\langle h(\llbracket \{B/V\} A \rrbracket^{\mathfrak{T}}|_k), \#_{\mu \oplus}(A) \rangle$ , where  $h : \mathfrak{T}^{fin} \rightarrow \mathbb{N}$  is the height function for finite trees and  $\#_{\mu \oplus}(A)$  is the number of occurrences of both  $\mu$  and  $\oplus$  at the head of  $A$ .

We proceed by analyzing the possible forms of  $A$  and assuming  $k > 0$  since the result for that case is immediate.

- $A = V$ : then  $\llbracket \{B/V\} V \rrbracket^{\mathfrak{T}}|_k = \llbracket B \rrbracket^{\mathfrak{T}}|_k = (\{\llbracket B \rrbracket^{\mathfrak{T}}/V\} V)|_k$  by Lem. 3.12.
- $A = a \neq V$ : then  $\llbracket \{B/V\} a \rrbracket^{\mathfrak{T}}|_k = \llbracket a \rrbracket^{\mathfrak{T}}|_k = a = (\{\llbracket B \rrbracket^{\mathfrak{T}}/V\} a)|_k$  by definition of the interpretation and Lem. 3.12.
- $A = D \textcircled{A} A'$ : then

$$\begin{aligned}
 \llbracket \{B/V\} A \rrbracket^{\mathfrak{T}}|_k &= \llbracket \{B/V\} D \textcircled{A} \{B/V\} A' \rrbracket^{\mathfrak{T}}|_k \\
 &= \llbracket \{B/V\} D \rrbracket^{\mathfrak{T}}|_{k-1} \textcircled{A} \llbracket \{B/V\} A' \rrbracket^{\mathfrak{T}}|_{k-1} && \text{by Def. 3.24 and 3.20} \\
 &= (\{\llbracket B \rrbracket^{\mathfrak{T}}/V\} \llbracket D \rrbracket^{\mathfrak{T}}|_{k-1}) \textcircled{A} (\{\llbracket B \rrbracket^{\mathfrak{T}}/V\} \llbracket A' \rrbracket^{\mathfrak{T}}|_{k-1}) && \text{by IH} \\
 &= (\{\llbracket B \rrbracket^{\mathfrak{T}}/V\} \llbracket D \rrbracket^{\mathfrak{T}} \textcircled{A} \{\llbracket B \rrbracket^{\mathfrak{T}}/V\} \llbracket A' \rrbracket^{\mathfrak{T}})|_k && \text{by Def. 3.20} \\
 &= (\{\llbracket B \rrbracket^{\mathfrak{T}}/V\} \llbracket D \textcircled{A} A' \rrbracket^{\mathfrak{T}})|_k && \text{by Lem. 3.12 and Def. 3.24}
 \end{aligned}$$

- $A = A' \supset A''$ : this case is similar to the previous one.
- $A = A_1 \oplus A_2$ : analysis for this case is similar to the previous ones but notice that we get the same  $k$  when resorting to Def. 3.20 (instead of  $k - 1$ ) before applying the inductive hypothesis. However, we are in conditions to apply it anyway since

$$h(\llbracket \{B/V\} A \rrbracket^{\mathfrak{T}}|_k) \geq h(\llbracket \{B/V\} A_i \rrbracket^{\mathfrak{T}}|_k) \quad \text{but} \quad \#_{\mu \oplus}(A) > \#_{\mu \oplus}(A_i)$$

Hence, it is safe to conclude  $\llbracket \{B/V\} A \rrbracket^{\mathfrak{T}}|_k = (\{\llbracket B \rrbracket^{\mathfrak{T}}/V\} \llbracket A \rrbracket^{\mathfrak{T}})|_k.$

- $A = \mu W.A'$ : without loss of generality we can assume  $\{B/V\}$  **avoids**  $W$ <sup>8</sup>. Then

$$\begin{aligned}
 \llbracket \{B/V\} A \rrbracket^{\mathfrak{T}}|_k &= \llbracket \mu W.\{B/V\} A' \rrbracket^{\mathfrak{T}}|_k \\
 &= \llbracket \{\mu W.\{B/V\} A'/W\} \{B/V\} A' \rrbracket^{\mathfrak{T}}|_k && \text{by Def. 3.24} \\
 &= \llbracket \{B/V\} \{A/W\} A' \rrbracket^{\mathfrak{T}}|_k \\
 &= (\llbracket B \rrbracket^{\mathfrak{T}}/V) \llbracket \{A/W\} A' \rrbracket^{\mathfrak{T}}|_k && \text{by IH} \\
 &= (\llbracket B \rrbracket^{\mathfrak{T}}/V) \llbracket A \rrbracket^{\mathfrak{T}}|_k && \text{by Def. 3.24}
 \end{aligned}$$

Here we are in condition to apply the inductive hypothesis since  $\#_{\mu\oplus}(A) > \#_{\mu\oplus}(\{A/W\} A')$  by contractiveness.  $\square$

The finite unfolding of a contractive  $\mu$ -type  $A$  consists of recursively replacing all occurrences of a bounded variable  $V$  by  $A$  itself a finite number of times. We formalize a slightly more general variation of this idea in the following lemma and prove its relation with  $\llbracket A \rrbracket^{\mathfrak{T}}$ .

**Lemma 3.26** *Let  $A = \mu V.A'$ ,  $B$  any other  $\mu$ -type and  $\sigma$  a substitution. Define*

$$A_{\sigma}^0 \triangleq B \quad A_{\sigma}^{n+1} \triangleq (\sigma \uplus \{A_{\sigma}^n/V\})A'$$

*Then,  $\forall k \in \mathbb{N}. \llbracket A_{\sigma}^k \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket \sigma A \rrbracket^{\mathfrak{T}}|_k$ .*

**Proof.** By induction on  $k$ . We assume without loss of generality that  $\sigma$  **avoids**  $V$ .

- $k = 0$ . Then  $\llbracket B \rrbracket^{\mathfrak{T}}|_0 = \bullet = \llbracket \sigma A \rrbracket^{\mathfrak{T}}|_0$  by definition of the truncation.
- $k > 0$ . By inductive hypothesis we have  $\llbracket A_{\sigma}^{k-1} \rrbracket^{\mathfrak{T}}|_{k-1} \simeq_{\mathfrak{T}} \llbracket \sigma A \rrbracket^{\mathfrak{T}}|_{k-1}$ . Moreover, since  $A = \mu V.A'$  is contractive, the first appearance of  $V$  in  $A'$  is at depth  $n > 1$ . So we have  $k \leq k-1+n$  and, by Lem. 3.13 and 3.25, we may conclude

$$\begin{aligned}
 \llbracket A_{\sigma}^k \rrbracket^{\mathfrak{T}}|_k &= \llbracket (\sigma \uplus \{A_{\sigma}^{k-1}/V\})A' \rrbracket^{\mathfrak{T}}|_k \\
 &= (\llbracket A_{\sigma}^{k-1} \rrbracket^{\mathfrak{T}}/V) \llbracket \sigma A' \rrbracket^{\mathfrak{T}}|_k && \text{by Lem. 3.25} \\
 &= (\llbracket A_{\sigma}^{k-1} \rrbracket^{\mathfrak{T}}|_{k-1}/V) \llbracket \sigma A' \rrbracket^{\mathfrak{T}}|_k && k \leq k-1+n \\
 &\simeq_{\mathfrak{T}} (\llbracket \sigma A \rrbracket^{\mathfrak{T}}|_{k-1}/V) \llbracket \sigma A' \rrbracket^{\mathfrak{T}}|_k && \text{by Lem. 3.13} \\
 &= (\llbracket \sigma A \rrbracket^{\mathfrak{T}}/V) \llbracket \sigma A' \rrbracket^{\mathfrak{T}}|_k && k \leq k-1+n \\
 &= \llbracket (\sigma \uplus \{\sigma A/V\})A' \rrbracket^{\mathfrak{T}}|_k && \text{by Lem. 3.25} \\
 &= \llbracket \sigma A \rrbracket^{\mathfrak{T}}|_k
 \end{aligned}$$

$\square$

**Remark 3.27** It follows immediately from the previous result that for every  $n \geq k$ ,  $\llbracket A_{\sigma}^n \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket \sigma A \rrbracket^{\mathfrak{T}}|_k$ .

<sup>8</sup> We use the predicate  $\sigma$  **avoids**  $V$  to mean that there is no collision at all between  $V$  and the variables in  $\sigma$  (i.e.  $V \notin \text{dom}(\sigma) \cap (\bigcup_{x \in \text{dom}(\sigma)} \text{fv}(\sigma x))$ ).

One of the main results of this section is the correspondence between the equivalence relations  $\simeq_\mu$  and  $\simeq_{\mathfrak{T}}$  via the function  $\llbracket \bullet \rrbracket^{\mathfrak{T}}$ . It follows from the lemma below that relates two  $\mu$ -equivalent types with the truncation of their respective trees:

**Lemma 3.28**  $A \simeq_\mu B$  iff  $\forall k \in \mathbb{N}. \llbracket A \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}}|_k$ .

**Proof.**  $\Rightarrow$ ) This part of the proof is by induction on  $A \simeq_\mu B$  analyzing the last rule applied. Note that  $\llbracket A \rrbracket^{\mathfrak{T}}|_0 = \bullet = \llbracket B \rrbracket^{\mathfrak{T}}|_0$  by definition of the truncation, so we only analyze the cases where  $k > 0$ .

- (E-REFL): then  $B = A$  and we conclude by reflexivity of  $\simeq_{\mathfrak{T}}$ ,  $\llbracket A \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket A \rrbracket^{\mathfrak{T}}|_k$  for every  $k > 0$ .
- (E-TRANS): then  $A \simeq_\mu C$  and  $C \simeq_\mu B$ . By inductive hypothesis  $\llbracket A \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket C \rrbracket^{\mathfrak{T}}|_k$  and  $\llbracket C \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}}|_k$  for every  $k > 0$ . Then we conclude by transitivity of  $\simeq_{\mathfrak{T}}$ .
- (E-SYMM): then  $B \simeq_\mu A$ . By inductive hypothesis  $\llbracket B \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket A \rrbracket^{\mathfrak{T}}|_k$  for every  $k > 0$  and we conclude by symmetry of  $\simeq_{\mathfrak{T}}$ .
- (E-FUNC): then  $A = A' \supset A'', B = B' \supset B''$  with  $A' \simeq_\mu B'$  and  $A'' \simeq_\mu B''$ . By inductive hypothesis  $\llbracket A' \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B' \rrbracket^{\mathfrak{T}}|_k$  and  $\llbracket A'' \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B'' \rrbracket^{\mathfrak{T}}|_k$  for every  $k > 0$ . Then

$$\llbracket A \rrbracket^{\mathfrak{T}}|_k = \llbracket A' \rrbracket^{\mathfrak{T}}|_{k-1} \supset \llbracket A'' \rrbracket^{\mathfrak{T}}|_{k-1} \simeq_{\mathfrak{T}} \llbracket B' \rrbracket^{\mathfrak{T}}|_{k-1} \supset \llbracket B'' \rrbracket^{\mathfrak{T}}|_{k-1} = \llbracket B \rrbracket^{\mathfrak{T}}|_k$$

- (E-COMP): then  $A = D @ A', B = D' @ B'$  with  $A' \simeq_\mu B'$  and  $A'' \simeq_\mu B''$ . This case is similar to the previous one. We conclude directly from the inductive hypothesis and the definition of the truncation

$$\llbracket D @ A' \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket D' @ B' \rrbracket^{\mathfrak{T}}|_k$$

- (E-UNION-IDEM): then  $A = B \oplus B$ . In this case we need to take into account that  $B$  may be a union type as well and, when working with  $\simeq_{\mathfrak{T}}$ , we must consider maximal union types. Let  $\llbracket A \rrbracket^{\mathfrak{T}}|_k = \bigoplus_{i \in 1..n} \mathcal{A}_i$  and  $\llbracket B \rrbracket^{\mathfrak{T}}|_k = \bigoplus_{j \in 1..m} \mathcal{B}_j$  with  $\mathcal{A}_j, \mathcal{B}_j \neq \oplus$ . It is immediate to see from the equality above that  $n = 2 * m$  and  $\mathcal{A}_j = \mathcal{A}_{2*j} = \mathcal{B}_j$  for every  $j \in 1..m$ . Finally we conclude by reflexivity of  $\simeq_{\mathfrak{T}}$  and (E-UNION-T)

$$\begin{aligned} \llbracket A \rrbracket^{\mathfrak{T}}|_k &= \bigoplus_{i \in 1..n} \mathcal{A}_i \\ &= (\bigoplus_{j \in 1..m} \mathcal{B}_j) \oplus (\bigoplus_{j \in 1..m} \mathcal{B}_j) \\ &\simeq_{\mathfrak{T}} \bigoplus_{j \in 1..m} \mathcal{B}_j \\ &= \llbracket B \rrbracket^{\mathfrak{T}}|_k \end{aligned}$$

- (E-UNION-COMM): then  $A = C_1 \oplus C_2$  and  $B = C_2 \oplus C_1$ . As in the previous case consider  $A|_k = \bigoplus_{i \in 1..n} \mathcal{A}_i$  and  $B|_k = \bigoplus_{j \in 1..m} \mathcal{B}_j$  with  $\mathcal{A}_i, \mathcal{B}_j \neq \oplus$ . Here  $n = m > 1$ , hence  $n + m > 2$ . Moreover, assuming  $\mathcal{A}_k$  is the last component of  $C_1$  ( $k \in 1..(n-1)$ ), we have  $\mathcal{A}_i = \mathcal{B}_{i+k}$  if  $i \leq n-k$ , and  $\mathcal{A}_i = \mathcal{B}_{i-(n-k)}$  if  $i > n-k$ . Thus, we conclude by reflexivity of  $\simeq_{\mathfrak{T}}$  and (E-UNION-T),  $\llbracket A \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}}|_k$ .

- (E-UNION-ASSOC): then  $A = C_1 \oplus (C_2 \oplus C_3)$  and  $B = (C_1 \oplus C_2) \oplus C_3$ . Considering maximal union types as before we have  $A|_k = \bigoplus_{i \in 1..n} \mathcal{A}_i$  and  $B|_k = \bigoplus_{j \in 1..m} \mathcal{B}_j$  with  $\mathcal{A}_i, \mathcal{B}_j \neq \oplus$  and  $n = m > 2$ . In this case we may conclude by resorting to the identity function in  $1..n$ , since  $\mathcal{A}_i = \mathcal{B}_i$ . Thus, by reflexivity and (E-UNION-T),  $\llbracket A \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}}|_k$ .
- (E-UNION): then  $A = A_1 \oplus A_2, B = B_1 \oplus B_2$  with  $A_1 \simeq_{\mu} B_1$  and  $A_2 \simeq_{\mu} B_2$ . By inductive hypothesis  $\llbracket A_1 \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B_1 \rrbracket^{\mathfrak{T}}|_k$  and  $\llbracket A_2 \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B_2 \rrbracket^{\mathfrak{T}}|_k$  for every  $k \in \mathbb{N}$ . Assume, without loss of generality

$$\begin{aligned} \llbracket A_1 \rrbracket^{\mathfrak{T}}|_k &= \bigoplus_{i \in 1..n} \mathcal{A}_i \quad \text{with} \quad \mathcal{A}_i \neq \oplus, i \in 1..n \\ \llbracket B_1 \rrbracket^{\mathfrak{T}}|_k &= \bigoplus_{j \in 1..m} \mathcal{B}_j \quad \text{with} \quad \mathcal{B}_j \neq \oplus, j \in 1..m \end{aligned}$$

If  $n + m > 2$ , there exists  $f : 1..n \rightarrow 1..m, g : 1..m \rightarrow 1..n$  such that  $\mathcal{A}_i \simeq_{\mathfrak{T}} \mathcal{B}_{f(i)}$  and  $\mathcal{A}_{g(j)} \simeq_{\mathfrak{T}} \mathcal{B}_j$ . If not (*i.e.*  $n = m = 1$ ), we simply take  $f = g = id$ .

Likewise, for  $A_2$  and  $B_2$  we have

$$\begin{aligned} \llbracket A_2 \rrbracket^{\mathfrak{T}}|_k &= \bigoplus_{i \in 1..n'} \mathcal{A}'_i \quad \text{with} \quad \mathcal{A}'_i \neq \oplus, i \in 1..n' \\ \llbracket B_2 \rrbracket^{\mathfrak{T}}|_k &= \bigoplus_{j \in 1..m'} \mathcal{B}'_j \quad \text{with} \quad \mathcal{B}'_j \neq \oplus, j \in 1..m' \end{aligned}$$

and there exists  $f' : 1..n' \rightarrow 1..m', g' : 1..m' \rightarrow 1..n'$  such that  $\mathcal{A}'_i \simeq_{\mathfrak{T}} \mathcal{B}'_{f'(i)}$  and  $\mathcal{A}'_{g'(j)} \simeq_{\mathfrak{T}} \mathcal{B}'_j$ .

Finally, since  $(n + n' + m + m') > 2$ , we can apply (E-UNION-T) to conclude

$$\begin{aligned} \llbracket A \rrbracket^{\mathfrak{T}}|_k &= \llbracket A_1 \rrbracket^{\mathfrak{T}}|_k \oplus \llbracket A_2 \rrbracket^{\mathfrak{T}}|_k \\ &= (\bigoplus_{i \in 1..n} \mathcal{A}_i) \oplus (\bigoplus_{i \in 1..n'} \mathcal{A}'_i) \\ &\simeq_{\mathfrak{T}} (\bigoplus_{j \in 1..m} \mathcal{B}_j) \oplus (\bigoplus_{j \in 1..m'} \mathcal{B}'_j) \\ &= \llbracket B_1 \rrbracket^{\mathfrak{T}}|_k \oplus \llbracket B_2 \rrbracket^{\mathfrak{T}}|_k \\ &= \llbracket B \rrbracket^{\mathfrak{T}}|_k \end{aligned}$$

- (E-REC): then  $A = \mu V. A', B = \mu V. B'$  with  $A' \simeq_{\mu} B'$ . By inductive hypothesis  $\llbracket A' \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B' \rrbracket^{\mathfrak{T}}|_k$  and, by Lem. 3.22,  $\llbracket A' \rrbracket^{\mathfrak{T}} \simeq_{\mathfrak{T}} \llbracket B' \rrbracket^{\mathfrak{T}}$ .

Now we consider the definition of  $A_{\sigma}^n$  and  $B_{\sigma}^n$  as in Lem. 3.26 with  $A_{\sigma}^0 \triangleq \sigma A'$  and  $B_{\sigma}^0 \triangleq \sigma B'$ . We claim that  $\llbracket A_{id}^n \rrbracket^{\mathfrak{T}} \simeq_{\mathfrak{T}} \llbracket B_{id}^n \rrbracket^{\mathfrak{T}}$  for every  $n \in \mathbb{N}$ . To prove this we proceed by induction on  $n$

- $n = 0$ . Then we have  $\llbracket A_{id}^0 \rrbracket^{\mathfrak{T}} = \llbracket A' \rrbracket^{\mathfrak{T}} \simeq_{\mathfrak{T}} \llbracket B' \rrbracket^{\mathfrak{T}} = \llbracket B_{id}^0 \rrbracket^{\mathfrak{T}}$  that holds by hypothesis.
- $n > 0$ . By reflexivity  $\{\llbracket A_{id}^{n-1} \rrbracket^{\mathfrak{T}}/V\} \llbracket A' \rrbracket^{\mathfrak{T}} \simeq_{\mathfrak{T}} \{\llbracket A_{id}^{n-1} \rrbracket^{\mathfrak{T}}/V\} \llbracket A' \rrbracket^{\mathfrak{T}}$ . Also, by inductive hypothesis,  $\llbracket A_{id}^{n-1} \rrbracket^{\mathfrak{T}} \simeq_{\mathfrak{T}} \llbracket B_{id}^{n-1} \rrbracket^{\mathfrak{T}}$  and, by hypothesis,  $\llbracket A' \rrbracket^{\mathfrak{T}} \simeq_{\mathfrak{T}} \llbracket B' \rrbracket^{\mathfrak{T}}$ . Then we can apply Lem. 3.13 and 3.25, and conclude

$$\llbracket A_{id}^n \rrbracket^{\mathfrak{T}} = \left\{ \llbracket A_{id}^{n-1} \rrbracket^{\mathfrak{T}}/V \right\} \llbracket A' \rrbracket^{\mathfrak{T}} \simeq_{\mathfrak{T}} \left\{ \llbracket B_{id}^{n-1} \rrbracket^{\mathfrak{T}}/V \right\} \llbracket B' \rrbracket^{\mathfrak{T}} = \llbracket B_{id}^n \rrbracket^{\mathfrak{T}}$$

Finally, by Lem. 3.22,  $\llbracket A_{id}^n \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B_{id}^n \rrbracket^{\mathfrak{T}}|_k$  for every  $k, n \in \mathbb{N}$ . Thus we conclude by Lem. 3.26

$$\llbracket A \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket A_{id}^k \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B_{id}^k \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}}|_k$$

- (E-FOLD): then  $A = \mu V.A'$  and  $B = \{\mu V.A'/V\} A'$ . The result is immediate by definition of the interpretation,  $\llbracket A \rrbracket^{\mathfrak{T}} = \llbracket \mu V.A' \rrbracket^{\mathfrak{T}} = \llbracket \{\mu V.A'/V\} A' \rrbracket^{\mathfrak{T}} = \llbracket B \rrbracket^{\mathfrak{T}}$ . Then  $\llbracket A \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}}|_k$  for every  $k \in \mathbb{N}$  by reflexivity of  $\simeq_{\mathfrak{T}}$ .
- (E-CONTR): then  $B = \mu V.B'$  is contractive and  $A \simeq_{\mu} \{A/V\} B'$ . By inductive hypothesis and Lem. 3.22,  $\llbracket A \rrbracket^{\mathfrak{T}} \simeq_{\mathfrak{T}} \llbracket \{A/V\} B' \rrbracket^{\mathfrak{T}}$ .

As in the previous case we consider  $B_{\sigma}^n$  from Lem. 3.26, this time with  $B_{\sigma}^0 \triangleq \sigma A$ . Now we show  $\llbracket A \rrbracket^{\mathfrak{T}} \simeq_{\mathfrak{T}} \llbracket B_{id}^n \rrbracket^{\mathfrak{T}}$  for every  $n \in \mathbb{N}$ , by induction on  $n$

- $n = 0$ . This case is immediate since  $\llbracket B_{id}^0 \rrbracket^{\mathfrak{T}} = \llbracket A \rrbracket^{\mathfrak{T}}$  by definition.
- $n > 0$ . Then, by definition and Lem. 3.25,  $\llbracket B_{id}^n \rrbracket^{\mathfrak{T}} = \{\llbracket B_{id}^{n-1} \rrbracket^{\mathfrak{T}}/V\} \llbracket B' \rrbracket^{\mathfrak{T}}$ . By inductive hypothesis we know  $\llbracket A \rrbracket^{\mathfrak{T}} \simeq_{\mathfrak{T}} \llbracket B_{id}^{n-1} \rrbracket^{\mathfrak{T}}$  and, by Lem. 3.13,  $\llbracket B_{id}^n \rrbracket^{\mathfrak{T}} \simeq_{\mathfrak{T}} \{\llbracket A \rrbracket^{\mathfrak{T}}/V\} \llbracket B' \rrbracket^{\mathfrak{T}}$ . Finally we conclude by applying Lem. 3.25 and transitivity of  $\simeq_{\mathfrak{T}}$  with hypothesis  $\llbracket A \rrbracket^{\mathfrak{T}} \simeq_{\mathfrak{T}} \llbracket \{A/V\} B' \rrbracket^{\mathfrak{T}}$

$$\llbracket B_{id}^n \rrbracket^{\mathfrak{T}} \simeq_{\mathfrak{T}} \llbracket \{A/V\} B' \rrbracket^{\mathfrak{T}} \simeq_{\mathfrak{T}} \llbracket A \rrbracket^{\mathfrak{T}}$$

Then, by Lem. 3.22,  $\llbracket A \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B_{id}^k \rrbracket^{\mathfrak{T}}|_k$  for every  $k, n \in \mathbb{N}$ . On the other hand, by Lem. 3.26, we know  $\llbracket B_{id}^k \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}}|_k$ . Thus, we conclude

$$\llbracket A \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B_{id}^k \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}}|_k$$

$\Leftarrow$ ) Let  $\llbracket A \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}}|_k$  for every  $k \in \mathbb{N}$ . Given  $B = \mu V.B'$  it is immediate to see that  $\llbracket \mu V.B' \rrbracket^{\mathfrak{T}} = \llbracket \{B/V\} B' \rrbracket^{\mathfrak{T}}$  while  $B \simeq_{\mu} \{B/V\} B'$ , by definition of the interpretation and (E-FOLD) respectively. Moreover, since  $\mu$ -types are contractive, we can assure that  $\#_{\mu}(\{B/V\} B') < \#_{\mu}(B)$ . By a simple induction on  $\#_{\mu}(B)$  we can prove that for every  $B \in \mathcal{T}$  there exists  $C \in \mathcal{T}$  such that  $\#_{\mu}(C) = 0$ ,  $B \simeq_{\mu} C$  and  $\llbracket B \rrbracket^{\mathfrak{T}} = \llbracket C \rrbracket^{\mathfrak{T}}$ . It is important to note that we are resorting to tree equality on this argument. Thus, without loss of generality, we consider during the proof only the cases where  $\#_{\mu}(B) = 0$ .

This proof is by induction on the lexicographical extension of the standard order to  $\langle h(\llbracket A \rrbracket^{\mathfrak{T}}|_k), \#_{\mu}(A) \rangle$ , where  $h : \mathfrak{T}^{fin} \rightarrow \mathbb{N}$  is the height function for finite trees. We proceed by analyzing the possible forms of  $A$ .

Given  $A, B \in \mathcal{T}$  we can assume

$$\begin{aligned} \llbracket A \rrbracket^{\mathfrak{T}} &= \bigoplus_{i \in 1..n} \mathcal{A}_i \quad \text{with} \quad \mathcal{A}_i \neq \bigoplus, i \in 1..n \\ \llbracket B \rrbracket^{\mathfrak{T}} &= \bigoplus_{j \in 1..m} \mathcal{B}_j \quad \text{with} \quad \mathcal{B}_j \neq \bigoplus, j \in 1..m \end{aligned}$$

by Rem. 3.7. Moreover, since  $\#_{\mu}(B) = 0$  and by definition of the interpretation, we have  $B = \bigoplus_{j \in 1..m} B_j$  with  $\llbracket B_j \rrbracket^{\mathfrak{T}} = \mathcal{B}_j$  for every  $j \in 1..m$  (note that  $B_j$  is a non-union type for every  $j \in 1..m$ ).

Then, we can divide this proof in two cases, either (i)  $A$  and  $B$  are both non-union types and thus  $n = m = 1$ ; or (ii) at least one of them is a union type (*i.e.*  $n + m > 2$ ).

- (i) If  $n = m = 1$ . Here we analyze the shape of  $A$ :
- $A = a$ . Then  $\llbracket A \rrbracket^{\mathfrak{T}}|_k = a$  for every  $k > 0$  and, by Lem. 3.10,  $\llbracket B \rrbracket^{\mathfrak{T}}|_k = \mathcal{B}_1|_k = a$ . Thus, by definition of the interpretation and tree truncation with the assumption  $\#_{\mu}(B) = 0$ , we have  $B = a$  and conclude with (E-REFL).
  - $A = D @ A'$ . Here we have  $\llbracket A \rrbracket^{\mathfrak{T}}|_k = \llbracket D \rrbracket^{\mathfrak{T}}|_{k-1} @ \llbracket A' \rrbracket^{\mathfrak{T}}|_{k-1}$  for every  $k > 0$  and, by Lem. 3.10 once again,  $\llbracket B \rrbracket^{\mathfrak{T}}|_k = \mathcal{B}'_k @ \mathcal{B}''_k$  with  $\llbracket D \rrbracket^{\mathfrak{T}}|_{k-1} \simeq_{\mathfrak{T}} \mathcal{B}'_k$  and  $\llbracket A' \rrbracket^{\mathfrak{T}}|_{k-1} \simeq_{\mathfrak{T}} \mathcal{B}''_k$ . With a similar analysis to the one made in Lem. 3.22, by definition of the interpretation and tree truncation with the assumption  $\#_{\mu}(B) = 0$ , we can assure that  $B = D' @ B'$  such that  $\mathcal{B}'_k = \llbracket D' \rrbracket^{\mathfrak{T}}|_{k-1}$  and  $\mathcal{B}''_k = \llbracket B' \rrbracket^{\mathfrak{T}}|_{k-1}$  for every  $k > 0$ . Then, we have  $\llbracket D \rrbracket^{\mathfrak{T}}|_{k-1} \simeq_{\mathfrak{T}} \llbracket D' \rrbracket^{\mathfrak{T}}|_{k-1}$  and  $\llbracket A' \rrbracket^{\mathfrak{T}}|_{k-1} \simeq_{\mathfrak{T}} \llbracket B' \rrbracket^{\mathfrak{T}}|_{k-1}$  and we can apply the inductive hypothesis to get  $D \simeq_{\mu} D'$  and  $A' \simeq_{\mu} B'$ . Finally we conclude by (E-COMP),  $D @ A' \simeq_{\mu} D' @ B'$ .
  - $A = A' \supset A''$ . Analysis for this case is similar to the previous one. From  $\llbracket A \rrbracket^{\mathfrak{T}}|_k = \llbracket A' \rrbracket^{\mathfrak{T}}|_{k-1} \supset \llbracket A'' \rrbracket^{\mathfrak{T}}|_{k-1}$  we get  $B = B' \supset B''$  with  $\llbracket A' \rrbracket^{\mathfrak{T}}|_{k-1} \simeq_{\mathfrak{T}} \llbracket B' \rrbracket^{\mathfrak{T}}|_{k-1}$  and  $\llbracket A'' \rrbracket^{\mathfrak{T}}|_{k-1} \simeq_{\mathfrak{T}} \llbracket B'' \rrbracket^{\mathfrak{T}}|_{k-1}$  for every  $k > 0$ . Then, by inductive hypothesis  $A' \simeq_{\mu} B'$  and  $A'' \simeq_{\mu} B''$ . Thus we conclude with (E-FUNC),  $A' \supset A'' \simeq_{\mu} B' \supset B''$ .
  - $A = \mu V.A'$  with  $A'$  a non-union type. By definition of the interpretation we have  $\llbracket A \rrbracket^{\mathfrak{T}}|_k = \llbracket \{A/V\} A' \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}}|_k$ . Here we may apply the inductive hypothesis as  $\#_{\mu}(\{A/V\} A') < \#_{\mu}(A)$ . Then,  $\{\mu V.A'/V\} A' \simeq_{\mu} B$ . On the other hand,  $\mu V.A' \simeq_{\mu} \{\mu V.A'/V\} A'$  by (E-FOLD). Finally we conclude with (E-TRANS),  $\mu V.A' \simeq_{\mu} B$ .
- (ii) If  $n + m > 2$ . Then the last rule applied to derive  $\llbracket A \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}}|_k$  is necessarily (E-UNION-T). Then, there exists  $f : 1..n \rightarrow 1..m, g : 1..m \rightarrow 1..n$  such that  $\mathcal{A}_i|_k \simeq_{\mathfrak{T}} \llbracket B_{f(i)} \rrbracket^{\mathfrak{T}}|_k$  and  $\mathcal{A}_{g(j)}|_k \simeq_{\mathfrak{T}} \llbracket B_j \rrbracket^{\mathfrak{T}}|_k$  for every  $i \in 1..n, j \in 1..m$ .
- If  $\#_{\mu}(A) \neq 0$ , then  $A = \mu V.A'$ ,  $\llbracket A \rrbracket^{\mathfrak{T}} = \llbracket \{A/V\} A' \rrbracket^{\mathfrak{T}}$  by definition and  $\#_{\mu}(\{A/V\} A') < \#_{\mu}(A)$  by contractivity. Thus we can conclude directly from the inductive hypothesis with (E-FOLD) and (E-TRANS) as before.
- If  $\#_{\mu}(A) = 0$ , by definition of the interpretation we have  $A = \bigoplus_{i \in 1..n} A_i$  with  $\llbracket A_i \rrbracket^{\mathfrak{T}} = \mathcal{A}_i$  for every  $i \in 1..n$ . Hence,  $\llbracket A_i \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B_{f(i)} \rrbracket^{\mathfrak{T}}|_k$  and  $\llbracket A_{g(j)} \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B_j \rrbracket^{\mathfrak{T}}|_k$ .
- Moreover, since  $\mathcal{A}_i, \mathcal{B}_j \neq \oplus$ , we are in the same situation as case (i) of this proof, so we can assure  $A_i \simeq_{\mu} B_{f(i)}$  and  $A_{g(j)} \simeq_{\mu} B_j$  for every  $i \in 1..n, j \in 1..m$ .
- Finally, we are under the hypothesis of Lem. 3.6, thus we conclude  $\bigoplus_{i \in 1..n} A_i \simeq_{\mu} \bigoplus_{j \in 1..m} B_j$ .

□

**Proposition 3.29**  $A \simeq_{\mu} B$  iff  $\llbracket A \rrbracket^{\mathfrak{T}} \simeq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}}$ .

**Proof.** This proposition follows from previous results shown on Lem. 3.22 and 3.28:  $A \simeq_{\mu} B$  iff  $\forall k \in \mathbb{N}. \llbracket A \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}}|_k$  iff  $\llbracket A \rrbracket^{\mathfrak{T}} \simeq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}}$ . □

To prove the correspondence between the subtyping relations we need to verify that all variable assumptions in the subtyping context can be substituted by convenient  $\mu$ -types before applying  $\llbracket \bullet \rrbracket^{\mathfrak{T}}$ .

**Lemma 3.30** *Let  $\Sigma = \{V_i \preceq_\mu W_i\}_{i \in 1..n}$  be a subtyping context and  $\sigma$  a substitution such that  $\text{dom}(\sigma) = \{V_i, W_i\}_{i \in 1..n}$ ,  $\sigma(V_i) = A_i$  and  $\sigma(W_i) = B_i$  with  $\text{dom}(\sigma) \cap \text{fv}(\{A_i, B_i\}_{i \in 1..n}) = \emptyset$ ,  $\llbracket A_i \rrbracket^\Sigma \preceq_\Sigma \llbracket B_i \rrbracket^\Sigma$  and  $A_i, B_i \in \mathcal{T}$  for every  $i \in 1..n$ . If  $\Sigma \vdash A \preceq_\mu B$ , then  $\llbracket \sigma A \rrbracket^\Sigma \preceq_\Sigma \llbracket \sigma B \rrbracket^\Sigma$ .*

**Proof.** By induction on  $\Sigma \vdash A \preceq_\mu B$  analyzing the last rule applied.

- (S-REFL):  $A = B$  and the result is immediate by reflexivity of  $\preceq_\Sigma$ .
- (S-TRANS):  $\Sigma \vdash A \preceq_\mu C$  and  $\Sigma \vdash C \preceq_\mu B$  for some  $C \in \mathcal{T}$ . By inductive hypothesis  $\llbracket \sigma A \rrbracket^\Sigma \preceq_\Sigma \llbracket \sigma C \rrbracket^\Sigma$  and  $\llbracket \sigma C \rrbracket^\Sigma \preceq_\Sigma \llbracket \sigma B \rrbracket^\Sigma$  for every  $\sigma$  satisfying the hypothesis of the lemma. Then we conclude by transitivity of  $\preceq_\Sigma$ .
- (S-HYP):  $A = V$  and  $B = W$  with  $\Sigma = \Sigma', V \preceq_\mu W$ . Then  $\sigma A = A_n$ ,  $\sigma B = B_n$  and the result is immediate since, by hypothesis of the lemma,  $\llbracket A_i \rrbracket^\Sigma \preceq_\Sigma \llbracket B_i \rrbracket^\Sigma$  for every  $i \in 1..n$ .
- (S-EQ):  $\vdash A \simeq_\mu B$  and, since  $\simeq_\mu$  is a congruence, we have  $\vdash \sigma A \simeq_\mu \sigma B$  for every substitution. So we can take  $\sigma$  satisfying the hypothesis of the lemma. Then, by Prop. 3.29,  $\llbracket \sigma A \rrbracket^\Sigma \simeq_\Sigma \llbracket \sigma B \rrbracket^\Sigma$  and we conclude by Lem. 3.19,  $\llbracket \sigma A \rrbracket^\Sigma \preceq_\Sigma \llbracket \sigma B \rrbracket^\Sigma$ .
- (S-FUNC):  $A = A' \supset A''$  and  $B = B' \supset B''$  with  $\Sigma \vdash B' \preceq_\mu A'$  and  $\Sigma \vdash A'' \preceq_\mu B''$ . By inductive hypothesis we have  $\llbracket \sigma B' \rrbracket^\Sigma \preceq_\Sigma \llbracket \sigma A' \rrbracket^\Sigma$  and  $\llbracket \sigma A'' \rrbracket^\Sigma \preceq_\Sigma \llbracket \sigma B'' \rrbracket^\Sigma$ . Then

$$\begin{aligned}
\llbracket \sigma A \rrbracket^\Sigma &= \llbracket \sigma A' \supset \sigma A'' \rrbracket^\Sigma \\
&= \llbracket \sigma A' \rrbracket^\Sigma \supset \llbracket \sigma A'' \rrbracket^\Sigma \\
&\preceq_\Sigma \llbracket \sigma B' \rrbracket^\Sigma \supset \llbracket \sigma B'' \rrbracket^\Sigma \\
&= \llbracket \sigma B' \supset \sigma B'' \rrbracket^\Sigma \\
&= \llbracket \sigma B \rrbracket^\Sigma
\end{aligned}$$

- (S-COMP):  $A = D @ A'$  and  $B = D' @ B'$  with  $\Sigma \vdash D \preceq_\mu D'$  and  $\Sigma \vdash A' \preceq_\mu B'$ . Similarly to the previous case we conclude from the inductive hypothesis that  $\llbracket \sigma D @ \sigma A' \rrbracket^\Sigma \preceq_\Sigma \llbracket \sigma D' @ \sigma B' \rrbracket^\Sigma$ .
- (S-UNION-L):  $A = A' \oplus A''$  with  $\Sigma \vdash A' \preceq_\mu B$  and  $\Sigma \vdash A'' \preceq_\mu B$ . By inductive hypothesis  $\llbracket \sigma A' \rrbracket^\Sigma \preceq_\Sigma \llbracket \sigma B \rrbracket^\Sigma$  and  $\llbracket \sigma A'' \rrbracket^\Sigma \preceq_\Sigma \llbracket \sigma B \rrbracket^\Sigma$ . Let

$$\begin{aligned}
\llbracket \sigma A' \rrbracket^\Sigma &= \bigoplus_{i \in 1..m} \mathcal{A}'_i & \mathcal{A}'_i &\neq \oplus \\
\llbracket \sigma A'' \rrbracket^\Sigma &= \bigoplus_{j \in 1..m'} \mathcal{A}''_j & \mathcal{A}''_j &\neq \oplus \\
\llbracket \sigma B \rrbracket^\Sigma &= \bigoplus_{k \in 1..l} \mathcal{B}_k & \mathcal{B}_k &\neq \oplus
\end{aligned}$$

Now we need to consider the following situations:

- (i)  $m = m' = l = 1$ . Then we conclude directly from the inductive hypothesis by applying (S-UNION-T),  $\llbracket \sigma A' \rrbracket^\Sigma \oplus \llbracket \sigma A'' \rrbracket^\Sigma = \mathcal{A}'_1 \oplus \mathcal{A}''_1 \preceq_\Sigma \mathcal{B}_1 = \llbracket \sigma B \rrbracket^\Sigma$ .
- (ii)  $m + l > 2$ . Then there exists  $f : 1..m \rightarrow 1..l$  such that  $\mathcal{A}'_i \preceq_\Sigma \mathcal{B}_{f(i)}$  and there are two possible cases:
  - (a)  $m' = l = 1$ . Then  $\mathcal{A}'_i \preceq_\Sigma \mathcal{B}_1$  (i.e.  $f$  is a constant function) and



$\mathcal{A}_1'' \preceq_{\mathfrak{T}} \mathcal{B}_1$ . Then we conclude by (S-UNION-T)

$$\llbracket \sigma A' \rrbracket^{\mathfrak{T}} \oplus \llbracket \sigma A'' \rrbracket^{\mathfrak{T}} = (\oplus_{i \in 1..m} \mathcal{A}'_i) \oplus \mathcal{A}_1'' \preceq_{\mathfrak{T}} \mathcal{B}_1 = \llbracket \sigma B \rrbracket^{\mathfrak{T}}$$

(b)  $m' + l > 2$ . Then there exists  $g : 1..m' \rightarrow 1..l$  such that  $\mathcal{A}_j'' \preceq_{\mathfrak{T}} \mathcal{B}_{g(j)}$ .  
Once again we conclude by (S-UNION-T)

$$\begin{aligned} \llbracket \sigma A' \rrbracket^{\mathfrak{T}} \oplus \llbracket \sigma A'' \rrbracket^{\mathfrak{T}} &= (\oplus_{i \in 1..m} \mathcal{A}'_i) \oplus (\oplus_{j \in 1..m'} \mathcal{A}''_j) \\ &\preceq_{\mathfrak{T}} \oplus_{k \in 1..l} \mathcal{B}_k \\ &= \llbracket \sigma B \rrbracket^{\mathfrak{T}} \end{aligned}$$

(iii) The only case left to analyze is  $m = l = 1$  and  $m' + l > 2$  that are similar to one where  $m' = l = 1$  and  $m + l > 2$ .

So we conclude that  $\llbracket \sigma A \rrbracket^{\mathfrak{T}} = \llbracket \sigma A' \rrbracket^{\mathfrak{T}} \oplus \llbracket \sigma A'' \rrbracket^{\mathfrak{T}} \preceq_{\mathfrak{T}} \llbracket \sigma B \rrbracket^{\mathfrak{T}}$ .

- (S-UNION-R1):  $B = B' \oplus B''$  with  $\Sigma \vdash A \preceq_{\mu} B'$ . By inductive hypothesis  $\llbracket \sigma A \rrbracket^{\mathfrak{T}} \preceq_{\mathfrak{T}} \llbracket \sigma B' \rrbracket^{\mathfrak{T}}$ . Let

$$\begin{aligned} \llbracket \sigma A \rrbracket^{\mathfrak{T}} &= \oplus_{i \in 1..m} \mathcal{A}_i & \mathcal{A}_i &\neq \oplus \\ \llbracket \sigma B' \rrbracket^{\mathfrak{T}} &= \oplus_{j \in 1..l} \mathcal{B}'_j & \mathcal{B}'_j &\neq \oplus \\ \llbracket \sigma B'' \rrbracket^{\mathfrak{T}} &= \oplus_{k \in 1..l'} \mathcal{B}''_k & \mathcal{B}''_k &\neq \oplus \end{aligned}$$

Here there are two possible situations:

(i)  $m = l = 1$ . Then  $\mathcal{A}_1 \preceq_{\mathfrak{T}} \mathcal{B}'_1$  and we conclude by (S-UNION-T)

$$\llbracket \sigma A \rrbracket^{\mathfrak{T}} = \mathcal{A}_1 \preceq_{\mathfrak{T}} \mathcal{B}'_1 \oplus (\oplus_{k \in 1..l'} \mathcal{B}''_k) = \llbracket \sigma B' \rrbracket^{\mathfrak{T}} \oplus \llbracket \sigma B'' \rrbracket^{\mathfrak{T}}$$

(ii)  $m + l > 2$ . Then there exists  $f : 1..m \rightarrow 1..l$  such that  $\mathcal{A}_i \preceq_{\mathfrak{T}} \mathcal{B}'_{f(i)}$ . We are again in a situation where all the conditions for (S-UNION-T) hold

$$\begin{aligned} \llbracket \sigma A \rrbracket^{\mathfrak{T}} &= \oplus_{i \in 1..m} \mathcal{A}_i \\ &\preceq_{\mathfrak{T}} (\oplus_{j \in 1..l} \mathcal{B}'_j) \oplus (\oplus_{k \in 1..l'} \mathcal{B}''_k) \\ &= \llbracket \sigma B' \rrbracket^{\mathfrak{T}} \oplus \llbracket \sigma B'' \rrbracket^{\mathfrak{T}} \end{aligned}$$

So we conclude that  $\llbracket \sigma A \rrbracket^{\mathfrak{T}} \preceq_{\mathfrak{T}} \llbracket \sigma B' \rrbracket^{\mathfrak{T}} \oplus \llbracket \sigma B'' \rrbracket^{\mathfrak{T}} = \llbracket \sigma B \rrbracket^{\mathfrak{T}}$ .

- (S-UNION-R2): this case is similar to the previous one, with  $B = B' \oplus B''$  and  $\Sigma \vdash A \preceq_{\mu} B''$ .
- (S-REC):  $A = \mu V. A', B = \mu W. B'$  with  $\Sigma, V \preceq_{\mu} W \vdash A' \preceq_{\mu} B', W \notin \text{fv}(A')$  and

$V \notin \text{fv}(B')$ . Let  $\sigma$  be a substitution satisfying the hypothesis of the lemma

- (1)  $\text{dom}(\sigma) = \{V_i, W_i\}_{i \in 1..n}$
- (2)  $\sigma(V_i) = A_i$  and  $\sigma(W_i) = B_i$
- (3)  $\{V_i, W_i\}_{i \in 1..n} \cap \text{fv}(\{A_i, B_i\}_{i \in 1..n}) = \emptyset$
- (4)  $\llbracket A_i \rrbracket^{\mathfrak{T}} \preceq_{\mathfrak{T}} \llbracket B_i \rrbracket^{\mathfrak{T}}$

Now consider  $A_\sigma^m$  and  $B_\sigma^m$  as in Lem. 3.26, recall

$$\begin{aligned} A_\sigma^0 &\triangleq \bullet & A_\sigma^{m+1} &\triangleq (\sigma \uplus \{A_\sigma^m/V\})A' \\ B_\sigma^0 &\triangleq \bullet & B_\sigma^{m+1} &\triangleq (\sigma \uplus \{B_\sigma^m/W\})B' \end{aligned}$$

and also the substitution  $\sigma_m = (\sigma \uplus \{A_\sigma^m/V\} \uplus \{B_\sigma^m/W\})$  for each  $m \in \mathbb{N}$ . Notice that  $\sigma_m A' = A_\sigma^{m+1}$  since  $W \notin \text{fv}(A')$ . Similarly,  $\sigma_m B' = B_\sigma^{m+1}$ .

It is immediate to see that  $\sigma_0$  satisfies the hypothesis of the lemma for the extended context  $\Sigma, V \preceq_\mu W$ , taking  $A_{n+1} = A_\sigma^0 = \bullet = B_\sigma^0 = B_{n+1}$ . This allows us to apply the inductive hypothesis and conclude that  $\llbracket A_\sigma^1 \rrbracket^{\mathfrak{T}} = \llbracket \sigma_0 A' \rrbracket^{\mathfrak{T}} \preceq_{\mathfrak{T}} \llbracket \sigma_0 B' \rrbracket^{\mathfrak{T}} = \llbracket B_\sigma^1 \rrbracket^{\mathfrak{T}}$ , and once again we are under the hypothesis of the lemma, this time with  $\sigma_1$ . Thus, directly from the inductive hypothesis (applied as many times as needed) we have  $\llbracket A_\sigma^m \rrbracket^{\mathfrak{T}} \preceq_{\mathfrak{T}} \llbracket B_\sigma^m \rrbracket^{\mathfrak{T}}$  for every  $m \in \mathbb{N}$ .

Then, by Lem. 3.23,  $\llbracket A_\sigma^m \rrbracket^{\mathfrak{T}}|_k \preceq_{\mathfrak{T}} \llbracket B_\sigma^m \rrbracket^{\mathfrak{T}}|_k$  for every  $k \in \mathbb{N}$ . Moreover, by Lem. 3.26 we have  $\llbracket \sigma A \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket A_\sigma^k \rrbracket^{\mathfrak{T}}|_k$  and  $\llbracket B_\sigma^k \rrbracket^{\mathfrak{T}}|_k \simeq_{\mathfrak{T}} \llbracket \sigma B \rrbracket^{\mathfrak{T}}|_k$ . Finally, by Lem. 3.19 and transitivity of subtyping we get  $\llbracket \sigma A \rrbracket^{\mathfrak{T}}|_k \preceq_{\mathfrak{T}} \llbracket \sigma B \rrbracket^{\mathfrak{T}}|_k$  and conclude with Lem. 3.23.  $\square$

Finally, as mentioned above, the following proposition and Lem. 3.17 allows us to prove Prop. 3.32.

**Proposition 3.31**  $A \preceq_\mu B$  iff  $\llbracket A \rrbracket^{\mathfrak{T}} \preceq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}}$ .

**Proof.**  $\Rightarrow$ ) This part of the proof follows directly from Lem. 3.30, taking  $\Sigma$  an empty subtyping context and thus  $\sigma$  results in the identity substitution. Hence from  $A \preceq_\mu B$  we get  $\llbracket A \rrbracket^{\mathfrak{T}} \preceq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}}$ .

$\Leftarrow$ ) For the converse we prove the equivalent result: if  $\forall k \in \mathbb{N}. \llbracket A \rrbracket^{\mathfrak{T}}|_k \preceq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}}|_k$  then,  $A \preceq_\mu B$ . And finally conclude by Lem. 3.23.

Let  $\llbracket A \rrbracket^{\mathfrak{T}}|_k \preceq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}}|_k$  for every  $k \in \mathbb{N}$ . As in the proof for Lem. 3.28, we only consider the cases where  $\#_\mu(B) = 0$  and proceed by induction on the lexicographical extension of the standard order to  $\langle h(\llbracket A \rrbracket^{\mathfrak{T}}|_k), \#_\mu(A) \rangle$ , analyzing the possible forms of  $A$ .

- $A = a$ . By definition of the interpretation and tree truncation we have  $\llbracket A \rrbracket^{\mathfrak{T}}|_k = a$  for every  $k > 0$ . Now, by definition of  $\preceq_{\mathfrak{T}}$ , only two rules apply:
  - (S-REFL-T): in this case we have  $\llbracket B \rrbracket^{\mathfrak{T}}|_k = a = B$ , by definition of the interpretation, and we conclude with (S-REFL).

- (S-UNION-T): by definition of the interpretation once again, we have  $B = \bigoplus_{i \in 1..n} B_i$  and

$$\llbracket a \rrbracket^{\mathfrak{T}}|_k \preceq_{\mathfrak{T}} \bigoplus_{i \in 1..n} \llbracket B_i \rrbracket^{\mathfrak{T}}|_k$$

with  $\llbracket a \rrbracket^{\mathfrak{T}}|_k \preceq_{\mathfrak{T}} \llbracket B_j \rrbracket^{\mathfrak{T}}|_k \neq \bigoplus$  for some  $j \in 1..n$ ,  $n > 1$ . Now the only applicable rule is (S-REFL-T), thus  $\llbracket B_j \rrbracket^{\mathfrak{T}}|_k = a = B_j$ . Then, by (S-REFL), (S-UNION-R1) and (S-UNION-R2), we conclude  $A \preceq_{\mu} \bigoplus_{i \in 1..n} B_i$ .

- $A = D @ A'$ . As before, by definition of the interpretation and tree truncation with  $k > 0$ ,  $\llbracket A \rrbracket^{\mathfrak{T}}|_k = \llbracket D \rrbracket^{\mathfrak{T}}|_{k-1} @ \llbracket A' \rrbracket^{\mathfrak{T}}|_{k-1} \preceq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}}|_k$ . The only two possible cases here are:
  - (S-COMP-T): by definition of the interpretation and tree truncation once again, we have  $B = D' @ B'$  with  $\llbracket D \rrbracket^{\mathfrak{T}}|_{k-1} \preceq_{\mathfrak{T}} \llbracket D' \rrbracket^{\mathfrak{T}}|_{k-1}$  and  $\llbracket A' \rrbracket^{\mathfrak{T}}|_{k-1} \preceq_{\mathfrak{T}} \llbracket B' \rrbracket^{\mathfrak{T}}|_{k-1}$ . Then, by inductive hypothesis,  $D \preceq_{\mu} D'$  and  $A' \preceq_{\mu} B'$ . Finally we conclude by (S-COMP),  $D @ A' \preceq_{\mu} D' @ B'$ .
  - (S-UNION-T): with a similar analysis as the case (S-UNION-T) for  $A = a$ , we have  $B = \bigoplus_{i \in 1..n} B_i$  and

$$\llbracket D @ A' \rrbracket^{\mathfrak{T}}|_k \preceq_{\mathfrak{T}} \bigoplus_{i \in 1..n} \llbracket B_i \rrbracket^{\mathfrak{T}}|_k$$

with  $\llbracket D @ A' \rrbracket^{\mathfrak{T}}|_k \preceq_{\mathfrak{T}} \llbracket B_j \rrbracket^{\mathfrak{T}}|_k \neq \bigoplus$  for some  $j \in 1..n$ ,  $n > 1$ . Then, by definition of  $\preceq_{\mathfrak{T}}$ , it is necessarily the case  $B_j = D' @ B'$  with  $\llbracket D \rrbracket^{\mathfrak{T}}|_k \preceq_{\mathfrak{T}} \llbracket D' \rrbracket^{\mathfrak{T}}|_k$  and  $\llbracket A' \rrbracket^{\mathfrak{T}}|_k \preceq_{\mathfrak{T}} \llbracket B' \rrbracket^{\mathfrak{T}}|_k$ . Now, as in the previous case, we have  $D @ A' \preceq_{\mu} B_j$  by inductive hypothesis. Finally, with (S-UNION-R1) and (S-UNION-R2), we conclude  $D @ A' \preceq_{\mu} \bigoplus_{i \in 1..n} B_i$ .

- $A = A' \supset A''$ . The only two applicable rules here are (S-FUNC-T) and (S-UNION-T). Both cases are similar to the ones exposed for  $@$ , concluding directly from the inductive hypothesis and the application of (S-FUNC) in the former while (S-UNION-R1) and (S-UNION-R2) are used in the latter.
- $A = \bigoplus_{i \in 1..n} A_i$  with  $A_i$  a non-union type for every  $i \in 1..n$ ,  $n > 1$ . This case is slightly simpler than the others as the only applicable rule is (S-UNION-T). Let  $B = \bigoplus_{j \in 1..m} B_j$  with  $B_j$  a non-union type for  $j \in 1..m$ . Note that  $m$  is not necessarily greater than 1. By definition of the interpretation and tree truncation we have, from (S-UNION-T),  $\exists f : 1..n \rightarrow 1..m$  such that  $\llbracket A_i \rrbracket^{\mathfrak{T}}|_k \preceq_{\mathfrak{T}} \llbracket B_{f(i)} \rrbracket^{\mathfrak{T}}|_k$  for every  $i \in 1..n$ . Then, by inductive hypothesis,  $A_i \preceq_{\mu} B_{f(i)}$  for every  $i \in 1..n$ . Now, by properly applying (S-UNION-R1) and (S-UNION-R2) on each case, we get  $A_i \preceq_{\mu} B$  for every  $i \in 1..n$ . Finally we conclude by multiple applications of (S-UNION-L),  $\bigoplus_{i \in 1..n} A_i$ .
- $A = \mu V.A'$ . Then  $\llbracket A \rrbracket^{\mathfrak{T}}|_k = \llbracket \{\mu V.A'/V\} A' \rrbracket^{\mathfrak{T}}|_k \preceq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}}|_k$ . By inductive hypothesis, with  $\#_{\mu}(\{A/V\} A') < \#_{\mu}(A)$ , we have  $\{\mu V.A'/V\} A' \preceq_{\mu} B$ . On the other hand, by (E-FOLD) and (S-EQ), we get  $\mu V.A' \preceq_{\mu} \{\mu V.A'/V\} A'$  and we conclude by (S-TRANS),  $\mu V.A' \preceq_{\mu} B$ .

□

**Proposition 3.32** (i) If  $D @ A \preceq_{\mu} D' @ A'$ , then  $D \preceq_{\mu} D'$  and  $A \preceq_{\mu} A'$ .

(ii) If  $A \supset B \preceq_{\mu} A' \supset B'$ , then  $A' \preceq_{\mu} A$  and  $B \preceq_{\mu} B'$ .

**Proof.** This result follows immediately from Lem. 3.17 and Prop. 3.31. □

### 3.1.5 Further properties on $\mu$ -types

We conclude the section with a simple but useful result on the preservation of the structure of non-union types by means of subtyping. Define the set of **union contexts** as the expressions generated by the following grammar

$$\mathcal{U} ::= \square \mid \mathcal{U} \oplus A \mid A \oplus \mathcal{U}$$

**Lemma 3.33** *For every type  $A \in \mathcal{T}$  there exists  $A' \in \mathcal{T}$  such that  $A \simeq_\mu A'$  and  $\#_\mu(A') = 0$ . Moreover, if  $\#_\mu(A) = 0$  then  $A$  and  $A'$  have the same outermost type constructor.*

**Proof.** By induction in  $\#_\mu(A)$ .

- $\#_\mu(A) = 0$ : the result is immediate taking  $A' = A$ . Notice that the second part of the statement holds trivially.
- $\#_\mu(A) > 0$ : then  $A = \mu V.A''$  and by rule (E-FOLD)  $A \simeq_\mu \{A/V\} A''$ . Since  $\mu$ -types are contractive we have  $\#_\mu(\{A/V\} A'') < \#_\mu(A)$ . Then, by inductive hypothesis, there exists  $A' \in \mathcal{T}$  such that  $A \simeq_\mu A'$ ,  $\#_\mu(A') = 0$ . Finally we conclude by rule (E-TRANS).

□

**Lemma 3.34** *If  $\mathcal{U}[A] \preceq_\mu B$  and  $A$  is a non-union type, then there exists a non-union type  $A' \in \mathcal{T}$  such that (i)  $B \simeq_\mu \mathcal{U}'[A']$ ; (ii)  $A \preceq_\mu A'$ ; and (iii)  $A$  and  $A'$  have the same outermost type constructor.*

**Proof.** By induction on the union context  $\mathcal{U}$ . Without loss of generality we can assume  $\#_\mu(A) = 0$ , by Lem. 3.33.

- $\mathcal{U} = \square$ . We have  $A \preceq_\mu B$ . By Prop. 3.31,  $\llbracket A \rrbracket^\mathfrak{T} \preceq_{\mathfrak{T}} \llbracket B \rrbracket^\mathfrak{T}$  where  $\llbracket A \rrbracket^\mathfrak{T} \neq \oplus$  by hypothesis. Let  $\llbracket B \rrbracket^\mathfrak{T} = \oplus_{i \in 1..n} \mathcal{B}_i$  with  $\mathcal{B}_i \neq \oplus$  for  $i \in 1..n$ . Note that  $\mathcal{B}_i$  is a subtree of the regular tree  $\llbracket B \rrbracket^\mathfrak{T}$ , thus it is regular too. Then, for every  $i \in 1..n$  there exists  $C_i \in \mathcal{T}$  such that  $\llbracket C_i \rrbracket^\mathfrak{T} = \mathcal{B}_i$ . Moreover, taking  $C = \oplus_{i \in 1..n} C_i$  we have  $\llbracket C \rrbracket^\mathfrak{T} = \llbracket B \rrbracket^\mathfrak{T}$ , hence  $C \simeq_\mu B$  by Prop. 3.29.
  - If  $n = 1$  (i.e.  $\llbracket B \rrbracket^\mathfrak{T} = \mathcal{B}_1 \neq \oplus$ ) the only applicable rules are (S-REFL-T), (S-FUNC-T) or (S-COMP-T), hence both trees have the same type constructor on the root. Applying Lem. 3.33 on  $B$  yields a type  $A'$  such that  $B \simeq_\mu A'$ , thus proving the first item with  $\mathcal{U}' = \square$ . This type  $A'$  has the same outermost type constructor as  $B$ , which we already saw is the same as  $A$ , hence proving item (iii). We are left to prove the second item. This follows from  $A \preceq_\mu B$ ,  $B \simeq_\mu A'$  by rules (E-TRANS) and (S-EQ).
  - If  $n > 1$ , then the only applicable rule is (S-UNION-T) and we have  $\llbracket A \rrbracket^\mathfrak{T} \preceq_{\mathfrak{T}} \llbracket C_j \rrbracket^\mathfrak{T} = \mathcal{B}_j \neq \oplus$  and, by Prop. 3.31,  $A \preceq_\mu C_j$  for some  $j \in 1..n$ . Note that both trees must have the same constructor in the root since neither of them is a union type (Lem. 3.17). Then we take the union context

$$\mathcal{U}' = C_1 \oplus \dots \oplus \square_j \oplus \dots \oplus C_n$$

and, by Lem. 3.33, there exists  $A' \in \mathcal{T}$  such that  $A' \simeq_\mu C_j$ ,  $\#_\mu(A') = 0$

and has the same outermost type constructor than  $C$ . Finally, we have

$$B \simeq_\mu C \simeq_\mu \mathcal{U}[A']$$

while  $A \preceq_\mu A'$  and both have the same outermost type constructor.

- $\mathcal{U} = C_1 \oplus \dots \oplus \square_k \oplus \dots \oplus C_m$  with  $m > 1$ , where  $C_k$  with  $k \in 1..m$  is the position of  $\square$  within  $\mathcal{U}$  (i.e.  $C_k = A$  in  $\mathcal{U}[A]$ ). We can assume without loss of generality that  $C_j$  is a non-union type for every  $j \in 1..m$ .

From  $\mathcal{U}[A] \preceq_\mu B$  and Prop. 3.31 we have  $\llbracket \mathcal{U}[A] \rrbracket^\mathfrak{T} \preceq_\mathfrak{T} \llbracket B \rrbracket^\mathfrak{T}$ . By definition

$$\llbracket \mathcal{U}[A] \rrbracket^\mathfrak{T} = \llbracket C_1 \rrbracket^\mathfrak{T} \oplus \dots \oplus \llbracket A \rrbracket^\mathfrak{T} \oplus \dots \oplus \llbracket C_m \rrbracket^\mathfrak{T}$$

with  $\llbracket C_j \rrbracket^\mathfrak{T} \neq \oplus$  for every  $j \in 1..m$ .

Assume once again  $\llbracket B \rrbracket^\mathfrak{T} = \bigoplus_{i \in 1..n} \mathcal{B}_i$  with  $\mathcal{B}_i \neq \oplus$  for  $i \in 1..n$ . The only subtyping rule that applies here is (S-UNION-T) since  $m > 1$ , hence  $n + m > 2$ . Then there exists  $f : 1..m \rightarrow 1..n$  such that  $\llbracket C_j \rrbracket^\mathfrak{T} \preceq_\mathfrak{T} \mathcal{B}_{f(j)}$  for every  $j \in 1..m$ .

Notice that  $\mathcal{U} = \mathcal{U}'' \oplus C_n$  or  $\mathcal{U} = C_1 \oplus \mathcal{U}''$  for some proper union context  $\mathcal{U}''$ . Hence, by construction

$$\mathcal{U}'' = C_1 \oplus \dots \oplus \square_k \oplus \dots \oplus C_{m-1} \quad \text{or} \quad \mathcal{U}'' = C_2 \oplus \dots \oplus \square_k \oplus \dots \oplus C_m$$

In either case, by rule (S-UNION-T), we have  $\llbracket \mathcal{U}''[A] \rrbracket^\mathfrak{T} \preceq_\mathfrak{T} \llbracket B \rrbracket^\mathfrak{T}$ , hence  $\mathcal{U}''[A] \preceq_\mu B$  by Prop. 3.31.

Finally, we can apply the inductive hypothesis to conclude that  $B \simeq_\mu \mathcal{U}'[A']$  with  $A' \in \mathcal{T}$  a non-union type such that  $A \preceq_\mu A'$  and both have the same outermost type constructor.

□

### 3.2 Typing Schemes

A **typing context**  $\Gamma$  (or  $\theta$ ) is a partial function from term variables to  $\mu$ -types;  $\Gamma(x) = A$  means that  $\Gamma$  maps  $x$  to  $A$ . We have two typing judgements, one for patterns  $\theta \vdash_p p : A$  and one for terms  $\Gamma \vdash s : A$ . Accordingly, we have two sets of typing rules: Fig. 5, top and bottom. We write  $\theta \triangleright_p p : A$  to indicate that the typing judgement  $\theta \vdash_p p : A$  is derivable (likewise for  $\Gamma \triangleright s : A$ ). The typing schemes speak for themselves except for two of them which we now comment. The first is (T-APP). Note that we do not require the  $A_i$  to be non-union types. This allows examples such as (5) to be typable (the outermost instance of (T-APP) is with  $n = 1$  and  $A_1 = \text{Bool} = \text{true} \oplus \text{false}$ ). Regarding (T-ABS) it requests a number of conditions. First of all, each of the patterns  $p_i$  must be typable under the typing context  $\theta_i$ ,  $i \in 1..n$ . Also, the set of free matchables in each  $p_i$  must be exactly the domain of  $\theta_i$ . Another condition, indicated by  $(\Gamma, \theta_i \vdash s_i : B)_{i \in 1..n}$ , is that the bodies of each of the branches  $s_i$ ,  $i \in 1..n$ , be typable under the context extended with the corresponding  $\theta_i$ . More noteworthy is the condition that the list  $[p_i : A_i]_{i \in 1..n}$  be *compatible*, which we now discuss in further detail.

## Patterns

$$\frac{\theta(x) = A}{\theta \vdash_{\mathbf{p}} x : A} \text{ (P-MATCH)} \quad \frac{}{\theta \vdash_{\mathbf{p}} \mathbf{c} : \mathbb{C}} \text{ (P-CONST)} \quad \frac{\theta \vdash_{\mathbf{p}} p : D \quad \theta \vdash_{\mathbf{p}} q : A}{\theta \vdash_{\mathbf{p}} pq : D @ A} \text{ (P-COMP)}$$

## Terms

$$\frac{\Gamma(x) = A}{\Gamma \vdash x : A} \text{ (T-VAR)} \quad \frac{}{\Gamma \vdash \mathbf{c} : \mathbb{C}} \text{ (T-CONST)} \quad \frac{\Gamma \vdash r : D \quad \Gamma \vdash u : A}{\Gamma \vdash ru : D @ A} \text{ (T-COMP)}$$

$$\frac{[p_i : A_i]_{i \in 1..n} \text{ compatible} \quad (\theta_i \vdash_{\mathbf{p}} p_i : A_i)_{i \in 1..n} \quad (\text{dom}(\theta_i) = \text{fm}(p_i))_{i \in 1..n} \quad (\Gamma, \theta_i \vdash s_i : B)_{i \in 1..n}}{\Gamma \vdash (p_i \rightarrow_{\theta_i} s_i)_{i \in 1..n} : \bigoplus_{i \in 1..n} A_i \supset B} \text{ (T-ABS)}$$

$$\frac{\Gamma \vdash r : \bigoplus_{i \in 1..n} A_i \supset B \quad \Gamma \vdash u : A_k \quad k \in 1..n}{\Gamma \vdash ru : B} \text{ (T-APP)} \quad \frac{\Gamma \vdash s : A \quad \vdash A \preceq_{\mu} A'}{\Gamma \vdash s : A'} \text{ (T-SUBS)}$$

Fig. 5. Typing rules for patterns and terms

### 3.3 Compatibility

Let us say that a pattern  $p$  **subsumes** a pattern  $q$ , written  $p \triangleleft q$  if there exists a substitution  $\sigma$  s.t.  $\sigma p = q$ . Consider an abstraction  $(p \rightarrow_{\theta} s \mid q \rightarrow_{\theta'} t)$  and two judgements  $\theta \vdash_{\mathbf{p}} p : A$  and  $\theta' \vdash_{\mathbf{p}} q : B$ . We consider two cases depending on whether  $p$  subsumes  $q$  or not.

As already mentioned in example (3) of the introduction, if  $p$  subsumes  $q$ , then the branch  $q \rightarrow_{\theta'} t$  will never be evaluated since the argument will already match  $p$ . Indeed, for any term  $u$  of type  $B$  in matchable form, the application will reduce to  $\llbracket u/p \rrbracket s$ . Thus, in this case, in order to ensure SR we demand that  $B \preceq_{\mu} A$ .

Suppose  $p$  does not subsume  $q$  (i.e.  $p \not\triangleleft q$ ). We analyze the cause of failure of subsumption in order to determine whether requirements on  $A$  and  $B$  must be put forward. In some cases no requirements are necessary. For example in:

$$\begin{aligned} f \rightarrow_{\{f:A \supset B\}} (\mathbf{c} \ z \rightarrow_{\{z:A\}} \mathbf{c} (f \ z)) \\ \mid \mathbf{d} \ y \rightarrow_{\{y:B\}} \mathbf{d} \ y \end{aligned} \tag{8}$$

no relation between  $A$  and  $B$  is required since the branches are mutually disjoint. In other cases, however,  $A \preceq_{\mu} B$  is required; we seek to characterize them. We focus on those cases where  $p$  fails to subsume  $q$ , and  $\pi \in \text{pos}(p) \cap \text{pos}(q)$  is an offending position in both patterns. The following table exhaustively lists them:

|     | $p _\pi$     | $q _\pi$     |                                   |
|-----|--------------|--------------|-----------------------------------|
| (a) |              | $y$          | restriction required              |
| (b) | $\mathbf{c}$ | $\mathbf{d}$ | no overlapping ( $q \not\leq p$ ) |
| (c) |              | $q_1 q_2$    | no overlapping                    |
| (d) |              | $y$          | restriction required              |
| (e) | $p_1 p_2$    | $\mathbf{d}$ | no overlapping                    |

In cases (b), (c) and (e), no extra condition on the types of  $p$  and  $q$  is necessary either, since their respective sets of possible arguments are disjoint; example (8) corresponds to the first of these. The cases where  $A$  and  $B$  must be related are (a) and (d): for those we require  $B \preceq_\mu A$ . The first of these has already been illustrated in the introduction (3), the second one is illustrated as follows:

$$f \rightarrow_{\{f:D \supset A \supset C\}} g \rightarrow_{\{g:B \supset C\}} (x y \rightarrow_{\{x:D, y:A\}} f x y \mid z \rightarrow_{\{z:B\}} g z) \quad (9)$$

The problematic situation is when  $B = D' @ B'$ , *i.e.* the type of  $z$  is another compound, which may have no relation at all with  $D @ A$ . Compatibility ensures  $B \preceq_\mu D @ A$ .

We now formalize these ideas.

**Definition 3.35** Given a pattern  $\theta \vdash_p p : A$  and  $\pi \in \text{pos}(p)$ , we say  $A$  admits a symbol  $\odot$  (with  $\odot \in \mathcal{V} \cup \mathcal{C} \cup \{\supset, @\}$ ) at position  $\pi$  iff  $\odot \in A|_\pi$ , where:

$$\begin{aligned} a|_\epsilon &\triangleq \{a\} \\ (A_1 \star A_2)|_\epsilon &\triangleq \{\star\}, & \star \in \{\supset, @\} \\ (A_1 \star A_2)|_{i\pi} &\triangleq A_i|_\pi, & \star \in \{\supset, @\}, i \in \{1, 2\} \\ (A_1 \oplus A_2)|_\pi &\triangleq A_1|_\pi \cup A_2|_\pi \\ (\mu V.A')|_\pi &\triangleq (\{\mu V.A'/V\} A')|_\pi \end{aligned}$$

Note that  $\theta \triangleright_p p : A$  and contractiveness of  $A$ , implies  $A|_\pi$  is well-defined for  $\pi \in \text{pos}(p)$ .

Whenever subsumption between two patterns fails, any mismatching position is a leaf in the syntactic tree of one of the patterns. Otherwise, both of them would have a type application constructor in that position and there would be no failure of subsumption.

**Definition 3.36** The *maximal positions* in a set of positions  $P$  are:

$$\text{maxpos}(P) \triangleq \{\pi \in P \mid \nexists \pi' \in P. \pi' = \pi \pi'' \wedge \pi'' \neq \epsilon\}$$



The *mismatching positions* between two patterns are:

$$\text{mmpos}(p, q) \triangleq \{\pi \mid \pi \in \text{maxpos}(\text{pos}(p) \cap \text{pos}(q)) \wedge p|_\pi \not\triangleleft q|_\pi\}$$

**Definition 3.37** We say  $p : A$  is *compatible* with  $q : B$ , written  $p : A \lll q : B$ , iff the following two conditions hold:

- (i)  $p \triangleleft q \implies B \preceq_\mu A$ .
- (ii)  $p \not\triangleleft q \implies (\forall \pi \in \text{mmpos}(p, q). A|_\pi \cap B|_\pi \neq \emptyset) \implies B \preceq A$ .

A list of patterns  $[p_i : A_i]_{i \in 1..n}$  is compatible if  $\forall i, j \in 1..n. i < j \implies p_i : A_i \lll p_j : A_j$ .

As a further example, suppose we wish to apply `upd` (cf. (1)) to data structures holding values of different types: say `v1` prefixed values are numbers and `v12` prefixed values are functions over numbers. Note that `upd` cannot be typed as it stands. The reason is that the last branch would have to handle values of functional type and hence would receive type  $\text{cons} \oplus \text{node} \oplus \text{nil} \oplus \text{v12} \oplus (\text{Nat} \supset \text{Nat})$ . This fails to be a datatype due to the presence of the component of functional type. As a consequence,  $x\ y$  cannot be typed since it requires an applicative type  $@$ . The remedy is to add an additional branch to `upd` capable of handling values prefixed by `v12`:

$$\begin{aligned} \text{upd}' = f \rightarrow_{\{f:A_1 \supset B\}} g \rightarrow_{\{g:(A_2 \supset A_3) \supset B\}} & \left( \begin{array}{ll} \text{v1 } z & \rightarrow_{\{z:A_1\}} \text{v1 } (f\ z) \\ \text{v12 } z & \rightarrow_{\{z:A_2 \supset A_3\}} \text{v12 } (gz) \\ x\ y & \rightarrow_{\{x:C, y:D\}} (\text{upd}'\ f\ x)\ (\text{upd}'\ f\ y) \\ w & \rightarrow_{\{w:E\}} w \end{array} \right) \end{aligned} \quad (10)$$

The type of `upd'` is  $(A_1 \supset B) \supset ((A_2 \supset A_3) \supset B) \supset (F_{A_1, A_2 \supset A_3} \supset F_{B, B})$ , where  $F_{X, Y}$  is

$$\mu\alpha. (\text{v1 } @\ X) \oplus (\text{v12 } @\ Y) \oplus (\alpha @\ \alpha) \oplus (\text{cons} \oplus \text{node} \oplus \text{nil})$$

This is quite natural: the type system establishes a clear distinction between semi-structured data, susceptible to path polymorphism, and “unstructured” data represented here by base and functional types.

### 3.4 Basic Metatheory of Typing

We present some technical lemmas that will be useful in the proof of safety and type-checking.

The following four lemmas are straightforward adaptations of the standard Generation Lemma and Basis Lemma to our system, considering patterns and terms separately.

**Lemma 3.38 (Generation Lemma for Patterns)** *Let  $\theta$  be a typing context and  $A$  a type.*

- (i) *If  $\theta \triangleright_p x : A$  then  $x : A \in \theta$ .*
- (ii) *If  $\theta \triangleright_p c : A$  then  $A \simeq_\mu c$ .*

(iii) If  $\theta \triangleright_p pq : A$  then  $\exists D, A'$  such that  $A \simeq_\mu D @ A'$ ,  $\theta \triangleright_p p : D$  and  $\theta \triangleright_p q : A'$ .

**Proof.** By simple analysis of the applicable rules for each term constructor. Note that here there's only one applicable rule in each case.  $\square$

**Lemma 3.39 (Basis Lemma for Patterns)** *Let  $\theta$  be a typing context,  $p$  a pattern and  $A$  a type such that  $\theta \triangleright_p p : A$ .*

(i) *Let  $\Delta \supseteq \theta$  be another typing context, then  $\theta \triangleright_p p : A$ .*

(ii)  *$\text{fm}(p) \subseteq \text{dom}(\theta)$ .*

(iii)  *$\theta|_{\text{fm}(p)} \triangleright_p p : A$ .*

**Proof.** The three cases are by induction on  $p$  using the Generation Lemma for Patterns.  $\square$

**Lemma 3.40 (Generation Lemma)** *Let  $\Gamma$  be a typing context and  $A$  a type.*

(i) *If  $\Gamma \triangleright x : A$  then  $\exists A'$  s.t.  $A' \preceq_\mu A$  and  $x : A' \in \Gamma$ .*

(ii) *If  $\Gamma \triangleright c : A$  then  $c \preceq_\mu A$ .*

(iii) *If  $\Gamma \triangleright ru : A$  then:*

(a) *either  $\exists D, A'$  s.t.  $D @ A' \preceq_\mu A$ ,  $\Gamma \triangleright r : D$  and  $\Gamma \triangleright u : A'$ ;*

(b) *or  $\exists A_1, \dots, A_n, A', k \in 1..n$  s.t.  $A' \preceq_\mu A$ ,  $\Gamma \triangleright r : \bigoplus_{i \in 1..n} A_i \supset A'$ , and  $\Gamma \triangleright u : A_k$ .*

(iv) *If  $\Gamma \triangleright (p_i \rightarrow_{\theta_i} s_i)_{i \in 1..n} : A$  then  $\exists A_1, \dots, A_n, B$  s.t.  $\bigoplus_{i \in 1..n} A_i \supset B \preceq_\mu A$ ,  $[p_i : A_i]_{i \in 1..n}$  is compatible,  $\text{dom}(\theta_i) = \text{fm}(p_i)$ ,  $\theta_i \triangleright_p p_i : A_i$  and  $\Gamma, \theta_i \triangleright s_i : B$  for every  $i \in 1..n$ .*

**Proof.** By induction on the derivation of  $\Gamma \vdash s : A$  analyzing the last rule applied.

- (T-VAR): then  $s = x$  with  $x : A' \in \Gamma$ . We take  $A = A'$  and (i) holds by reflexivity of subtyping.
- (T-CONST): then  $s = c$  and  $A = c$ . Again by reflexivity we conclude that (ii) holds.
- (T-COMP): then  $s = ru$  and  $A = D @ A'$  with  $\Gamma \triangleright r : D$  and  $\Gamma \triangleright u : A'$ . By reflexivity of subtyping we get  $D @ A' \preceq_\mu A$  and conclude that (iii.a) holds.
- (T-ABS): then  $s = (p_i \rightarrow_{\theta_i} s_i)_{i \in 1..n}$  and  $A = \bigoplus_{i \in 1..n} A_i \supset B$  with  $\text{dom}(\theta_i) = \text{fm}(p_i)$ ,  $[p_i : A_i]_{i \in 1..n}$  compatible,  $\theta_i \triangleright_p p_i : A_i$  and  $\Gamma, \theta_i \triangleright s_i : B$  for every  $i \in 1..n$ . Here (iv) holds by reflexivity of subtyping.
- (T-APP): then  $s = ru$  with  $\Gamma \triangleright r : \bigoplus_{i \in 1..n} A_i \supset A$  and  $\Gamma \triangleright u : A_k$  for some  $k \in 1..n$ . We conclude reflexivity with  $A' = A$  that (iii.b) holds.
- (T-SUBS): then  $\Gamma \triangleright s : A''$  with  $A'' \preceq_\mu A$ . Now we analyze the form of the term  $s$  to see which of the cases of the lemma holds for each term constructor:
  - (i)  $s = x$ . By inductive hypothesis  $\exists A'$  such that  $A' \preceq_\mu A''$  and  $x : A' \in \Gamma$ . Then, by transitivity of subtyping,  $A' \preceq_\mu A''$  and we conclude that (i) holds.
  - (ii)  $s = c$ . By inductive hypothesis  $c \preceq_\mu A''$  and by transitivity of subtyping  $c \preceq_\mu A$ . Hence (ii) holds.

- (iii)  $s = r u$ . By inductive hypothesis we have two options:
- (a) either  $\exists D, A'$  such that  $D @ A' \preceq_\mu A'', \Gamma \triangleright r : D$  and  $\Gamma \triangleright u : A'$ . By transitivity we have  $D @ A' \preceq_\mu A$  and we are in the case that (iii.a) holds.
  - (b) or  $\exists A_1, \dots, A_n, A'$  such that  $A \preceq_\mu A'', \Gamma \triangleright r : \bigoplus_{i \in 1..n} A_i \supset A'$ , and  $\Gamma \triangleright u : A_k$  for some  $k \in 1..n$ . Again by transitivity  $A' \preceq_\mu A$  and (iii.b) holds.
- (iv)  $s = (p_i \rightarrow_{\theta_i} s_i)_{i \in 1..n}$ . By inductive hypothesis  $\exists A_1, \dots, A_n, B$  such that  $\bigoplus_{i \in 1..n} A_i \supset B \preceq_\mu A'', [p_i : A_i]_{i \in 1..n}$  is compatible,  $\text{dom}(\theta_i) = \text{fm}(p_i)$ ,  $\theta_i \triangleright_p p_i : A_i$  and  $\Gamma, \theta_i \triangleright s_i : B$  for every  $i \in 1..n$ . Then we conclude by transitivity of subtyping that  $\bigoplus_{i \in 1..n} A_i \supset B \preceq_\mu A$  and (iv) holds.  $\square$

**Lemma 3.41 (Basis Lemma)** *Let  $\Gamma$  be a typing context,  $s$  a term and  $A$  a type such that  $\Gamma \triangleright s : A$ .*

- (i) *Let  $\Delta \supseteq \Gamma$  be another typing context, then  $\Delta \triangleright s : A$ .*
- (ii)  *$\text{fv}(s) \subseteq \text{dom}(\Gamma)$ .*
- (iii)  *$\Gamma|_{\text{fv}(s)} \triangleright s : A$ .*

**Proof.** The three cases are by induction on  $s$  using the Generation Lemma.

- (i)  $\Delta \triangleright s : A$ .
  - $s = x$ . By Lem. 3.40 (i)  $\exists A'$  such that  $A' \preceq_\mu A$  and  $x : A' \in \Gamma$ . Then  $\Delta = \Delta', x : A'$  and by (T-VAR) and (T-SUBS) we get  $\Delta \triangleright x : A$ .
  - $s = c$ . By Lem. 3.40 (ii)  $c \preceq_\mu A$  and we conclude by (T-CONST) and (T-SUBS)  $\Delta \triangleright c : A$ .
  - $s = r u$ . By Lem. 3.40 (iii) we have two possible cases:
    - (a) either  $\exists D, A'$  such that  $D @ A' \preceq_\mu A, \Gamma \triangleright r : D$  and  $\Gamma \triangleright u : A'$ . By inductive hypothesis  $\Delta \triangleright r : D$  and  $\Delta \triangleright u : A'$ . Then by (T-COMP) and (T-SUBS)  $\Delta \triangleright r u : A$ .
    - (b) or  $\exists A_1, \dots, A_n, A'$  such that  $A' \preceq_\mu A, \Gamma \triangleright r : \bigoplus_{i \in 1..n} A_i \supset A'$ , and  $\Gamma \triangleright u : A_k$  for some  $k \in 1..n$ . Applying the inductive hypothesis we get  $\Delta \triangleright r : \bigoplus_{i \in 1..n} A_i \supset A'$  and  $\Delta \triangleright u : A_k$ , so we conclude by (T-APP) and (T-SUBS) that  $\Delta \triangleright r u : A$ .
  - $s = (p_i \rightarrow_{\theta_i} s_i)_{i \in 1..n}$ . By Lem. 3.40 (iv)  $\exists A_1, \dots, A_n, B$  such that  $\bigoplus_{i \in 1..n} A_i \supset B \preceq_\mu A, [p_i : A_i]_{i \in 1..n}$  is compatible,  $\text{dom}(\theta_i) = \text{fm}(p_i)$ ,  $\theta_i \triangleright_p p_i : A_i$  and  $\Gamma, \theta_i \triangleright s_i : B$  for every  $i \in 1..n$ . Without loss of generality we can assume  $\text{dom}(\Delta) \cap \text{dom}(\theta_i) = \emptyset$  for all  $i \in 1..n$ . Then  $\Delta, \theta_i$  is also a typing context and by inductive hypothesis  $\Delta, \theta_i \triangleright s_i : B$  for all  $i \in 1..n$ . Then by (T-ABS) and (T-SUBS) we conclude  $\Delta \triangleright (p_i \rightarrow_{\theta_i} s_i)_{i \in 1..n} : A$ .
- (ii)  $\text{fv}(s) \subseteq \text{dom}(\Gamma)$ .
  - $s = x$ . By Lem. 3.40 (i)  $\exists A'$  such that  $A' \preceq_\mu A$  and  $x : A' \in \Gamma$ . Then  $\text{fv}(s) = \{x\} \subseteq \text{dom}(\Gamma)$ .
  - $s = c$ . Then  $\text{fv}(s) = \emptyset \subseteq \text{dom}(\Gamma)$ .
  - $s = r u$ . By Lem. 3.40 (iii)  $\exists B, B'$  such that  $\Gamma \triangleright r : B$  and  $\Gamma \triangleright u : B'$ . By inductive hypothesis  $\text{fv}(r) \subseteq \text{dom}(\Gamma)$  and  $\text{fv}(u) \subseteq \text{dom}(\Gamma)$ . Then  $\text{fv}(s) = \text{fv}(r) \cup \text{fv}(u) \subseteq \text{dom}(\Gamma)$ .

- $s = (p_i \rightarrow_{\theta_i} s_i)_{i \in 1..n}$ . By Lem. 3.40 (iv)  $\exists A_1, \dots, A_n, B$  such that  $\bigoplus_{i \in 1..n} A_i \supset B \preceq_{\mu} A$ ,  $\text{dom}(\theta_i) = \text{fm}(p_i)$  and  $\Gamma, \theta_i \triangleright s_i : B$  for every  $i \in 1..n$ . By inductive hypothesis  $\text{fv}(s_i) \subseteq \text{dom}(\Gamma, \theta_i) = \text{dom}(\Gamma) \uplus \text{fm}(p_i)$  and we have  $\text{fv}(s_i) \setminus \text{fm}(p_i) \subseteq \text{dom}(\Gamma)$  for every  $i \in 1..n$ . Then  $\text{fv}(s) = \bigcup_{i \in 1..n} \text{fv}(s_i) \setminus \text{fm}(p_i) \subseteq \text{dom}(\Gamma)$ .
- (iii)  $\Gamma|_{\text{fv}(s)} \triangleright s : A$ .
  - $s = x$ . By Lem. 3.40 (i)  $\exists A'$  such that  $A' \preceq_{\mu} A$  and  $x : A' \in \Gamma$ . Then by (T-VAR) and (T-SUBS)  $\Gamma|_{\text{fv}(s)} = x : A' \triangleright x : A$ .
  - $s = c$ . By Lem. 3.40 (ii)  $c \preceq_{\mu} A$  and we conclude by (T-CONST) and (T-SUBS)  $\Gamma|_{\text{fv}(s)} \triangleright c : A$ .
  - $s = r u$ . By Lem. 3.40 (iii) we have two possible cases:
    - (a) either  $\exists D, A'$  such that  $D @ A' \preceq_{\mu} A$ ,  $\Gamma \triangleright r : D$  and  $\Gamma \triangleright u : A'$ . By inductive hypothesis  $\Gamma|_{\text{fv}(r)} \triangleright r : D$  and  $\Gamma|_{\text{fv}(u)} \triangleright u : A'$ . Since  $\Gamma$  is a typing context,  $\Gamma|_{\text{fv}(r)} \subseteq \Gamma$  and  $\Gamma|_{\text{fv}(u)} \subseteq \Gamma$ , then  $\Gamma|_{\text{fv}(r)} \cup \Gamma|_{\text{fv}(u)} = \Gamma|_{\text{fv}(ru)}$  is also a typing context. Now, by Lem. 3.41 (i),  $\Gamma|_{\text{fv}(s)} \triangleright r : D$  and  $\Gamma|_{\text{fv}(s)} \triangleright u : A'$ . Then we conclude by applying (T-COMP) and (T-SUBS).
    - (b) or  $\exists A_1, \dots, A_n, A'$  such that  $A' \preceq_{\mu} A$ ,  $\Gamma \triangleright r : \bigoplus_{i \in 1..n} A_i \supset A'$ , and  $\Gamma \triangleright u : A_k$  for some  $k \in 1..n$ . By inductive hypothesis  $\Gamma|_{\text{fv}(u)} \triangleright r : \bigoplus_{i \in 1..n} A_i \supset A'$  and  $\Gamma|_{\text{fv}(u)} \triangleright u : A_k$ . Again we have  $\Gamma|_{\text{fv}(r)} \cup \Gamma|_{\text{fv}(u)} = \Gamma|_{\text{fv}(ru)}$  a typing context and we can apply case (i) of this same lemma to get  $\Gamma|_{\text{fv}(s)} \triangleright r : \bigoplus_{i \in 1..n} A_i \supset A'$  and  $\Gamma|_{\text{fv}(s)} \triangleright u : A_k$ . Finally we conclude by (T-APP) and (T-SUBS).
  - $s = (p_i \rightarrow_{\theta_i} s_i)_{i \in 1..n}$ . By Lem. 3.40 (iv)  $\exists A_1, \dots, A_n, B$  such that  $\bigoplus_{i \in 1..n} A_i \supset B \preceq_{\mu} A$ ,  $[p_i : A_i]_{i \in 1..n}$  is compatible,  $\text{dom}(\theta_i) = \text{fm}(p_i)$ ,  $\theta_i \triangleright_p p_i : A_i$  and  $\Gamma, \theta_i \triangleright s_i : B$  for every  $i \in 1..n$ . By inductive hypothesis  $(\Gamma, \theta_i)|_{\text{fv}(s_i)} \triangleright s_i : B$  and it's immediate to see that  $(\Gamma, \theta_i)|_{\text{fv}(s_i)} \subseteq \Gamma|_{\text{fv}(s_i)} \uplus \theta_i$ , since  $\text{dom}(\Gamma) \cap \text{dom}(\theta_i) = \emptyset$ . Moreover, as  $\text{dom}(\theta_i) = \text{fm}(p_i)$ , we have  $\Gamma|_{\text{fv}(s_i)} = \Gamma|_{(\text{fv}(s_i) \setminus \text{fm}(p_i))} \subseteq \Gamma|_{\bigcup_{j \in 1..n} (\text{fv}(s_j) \setminus \text{fm}(p_j))}$ . Then  $\Gamma|_{\text{fv}(s)} \uplus \theta_i \supseteq \Gamma|_{\text{fv}(s_i)} \uplus \theta_i \supseteq (\Gamma, \theta_i)|_{\text{fv}(s_i)}$  is also a typing context and, by Lem. 3.41 (i), we get  $\Gamma|_{\text{fv}(s)}, \theta_i \triangleright s_i : B$  for every  $i \in 1..n$ . Finally we apply (T-ABS) and (T-SUBS) to conclude  $\Gamma|_{\text{fv}(s)} \triangleright (p_i \rightarrow_{\theta_i} s_i)_{i \in 1..n} : A$ .

□

The following lemma is useful to deduce the shape of the type when we know the term is a data structure. Essentially it states that every data structure that can be given a type, can also be typed with a more specific non-union datatype.

**Lemma 3.42 (Typing for Data Structures)** *Suppose  $\Gamma \triangleright d : A$ , for  $d$  a data structure. Then  $\exists D$  datatype such that  $D$  is a non-union type,  $D \preceq_{\mu} A$  and  $\Gamma \triangleright d : D$ . Moreover,*

- (i) *If  $d = c$ , then  $D \simeq_{\mu} c$ .*
- (ii) *If  $d = d' t$ , then  $\exists D', A'$  such that  $D \simeq_{\mu} D' @ A'$ ,  $\Gamma \triangleright d' : D'$  and  $\Gamma \triangleright t : A'$ .*

**Proof.** By induction on  $d$ .

- $d = c$ . By Lem. 3.40 (ii)  $D = c \preceq_{\mu} A$ .
- $d = d' t$ . By Lem. 3.40 (iii) there are two possible cases:

- (a) either  $\exists D', A'$  such that  $D' @ A' \preceq_\mu A$ ,  $\Gamma \triangleright d' : D'$  and  $\Gamma \triangleright t : A'$ . Then the property holds with  $D = D' @ A'$ , since by (T-COMP) we can derive  $\Gamma \triangleright d' t : D$ .
- (b) or  $\exists A_1, \dots, A_n, A'$  such that  $A' \preceq_\mu A$ ,  $\Gamma \triangleright d' : \bigoplus_{i \in 1..n} A_i \supset A'$ , and  $\Gamma \triangleright t : A_k$  for some  $k \in 1..n$ . By inductive hypothesis applied to  $d'$  we get that  $\exists D'$  datatype such that  $D'$  is not a union type and  $D' \preceq_\mu \bigoplus_{i \in 1..n} A_i \supset A'$ . But, by Lem. 3.34, both of them have the same outermost type constructor, which leads to a contradiction. Hence this case does not apply.

□

Some results on compatibility follow, the crucial one being Lem. 3.44. This next lemma shows that matching failure is enough to guarantee that the type of the argument is not a subtype of that of the pattern.

**Lemma 3.43** *Given  $\Gamma \triangleright u : B$ ,  $\theta \triangleright_p p : A$ . If  $\llbracket u/p \rrbracket = \text{fail}$ , then  $B \not\preceq_\mu A$ .*

**Proof.** By induction on  $p$ . We only analyse the cases where  $\llbracket u/p \rrbracket = \text{fail}$ , otherwise the implication holds trivially.

- $p = c$ : then  $u$  is a matchable form and  $u \neq c$ . By Lem. 3.38 (ii),  $A = c$ .
  - (i)  $u = d \neq c$ : by Lem. 3.40 (ii),  $d \preceq_\mu B$ . Then, if  $B \preceq_\mu c$  we would have  $d \preceq_\mu c$  by transitivity, which is clearly not possible by invertibility of subtyping for non-union types. Hence, it cannot be the case that  $B \preceq_\mu A$ .
  - (ii)  $u = u_1 u_2$ : by Lem. 3.42,  $\exists D', B'$  such that  $D' @ B' \preceq_\mu B$ . Again, if  $B \preceq_\mu c$  we have a contradiction.
  - (iii)  $u = (q_j \rightarrow u_j)_{j \in 1..m}$ : by Lem. 3.40 (iv), there exists  $B_1, \dots, B_m, B'$  such that  $\bigoplus_{j \in 1..m} B_j \supset B' \preceq_\mu B$ . Thus, we conclude by contradiction as in the previous case.
- $p = p_1 p_2$ : here, by Lem. 3.38 (iii),  $\exists D, A'$  such that  $A = D @ A'$  with  $\theta \triangleright_p p_1 : D$  and  $\theta \triangleright_p p_2 : A'$ . There are three possible cases of mismatch:
  - (i)  $u = d \neq c$ : similarly to the previous cases, by Lem. 3.40 (ii) we have  $d \preceq_\mu B$  which leads to a contradiction if  $B \preceq_\mu D @ A'$ .
  - (ii)  $u = u_1 u_2$ : then the mismatch was internal. Thus, we have  $\llbracket u_i/p_i \rrbracket = \text{fail}$  for at least one of the two possibilities. By Lem. 3.42,  $\exists D', B'$  such that  $D' @ B' \preceq_\mu B$  with  $\Gamma \triangleright u_1 : D'$  and  $\Gamma \triangleright u_2 : B'$ . Then, by inductive hypothesis, we have  $D \not\preceq_\mu D'$ , or  $A' \not\preceq_\mu B'$ , or both.
 

Now suppose  $B \preceq_\mu A \simeq_\mu D' @ A'$ . By transitivity we have  $D' @ B' \preceq_\mu D' @ A'$  and by invertibility of subtyping for non-union types both  $D \preceq_\mu D'$  and  $A' \preceq_\mu B'$  should hold. Thus, we conclude  $B \not\preceq_\mu A$ .
  - (iii)  $u = (q_j \rightarrow u_j)_{j \in 1..m}$ : as before, by Lem. 3.40 (iv), we have  $B_1, \dots, B_m, B'$  such that  $\bigoplus_{j \in 1..m} B_j \supset B' \preceq_\mu B$  and conclude by contradiction with  $B \preceq_\mu D @ A'$ .

□

Define  $\mathcal{P}_{\text{comp}}(p : A, q : B) \triangleq \forall \pi \in \text{mmpos}(p, q). A \parallel_\pi \cap B \parallel_\pi \neq \emptyset$ , so that compatibility can alternatively be characterized as:

$$p : A \lll q : B \quad \text{iff} \quad \mathcal{P}_{\text{comp}}(p : A, q : B) \implies B \preceq_\mu A$$

The Compatibility Lemma should be interpreted in the context of an abstraction. Assume an argument  $u$  of type  $B$  is passed to a function where there are (at least) two branches, defined by patterns  $p$  and  $q$ , the latter having the same type as  $u$ . If the argument matches the first pattern of (potentially) a different type  $A$ , then  $\mathcal{P}_{\text{comp}}(p : A, q : B)$  must hold. Since patterns within an abstraction must be compatible, we get  $B \preceq_\mu A$  and thus  $\Gamma \triangleright u : A$  too.

**Lemma 3.44 (Compatibility Lemma)** *Suppose  $\Gamma \triangleright u : B$ ,  $\theta \triangleright_p p : A$ ,  $\theta' \triangleright_p q : B$  and  $\llbracket u/p \rrbracket$  is successful. Then,  $\mathcal{P}_{\text{comp}}(p : A, q : B)$  holds.*

**Proof.** By induction on  $p$ .

- $p = x$ : then the result is immediate since  $x \triangleleft q$  for every pattern  $q$ .
- $p = c$ : if  $c \triangleleft q$  the result is immediate. So let's analyze the case where  $c \not\triangleleft q$  (i.e.  $q \neq c$ ). We have  $u = c$  by matching success and  $c \preceq_\mu B$  by Lem. 3.40 (ii). Assume  $B = \bigoplus_{i \in 1..n} B_i$  with  $B_i \neq \oplus$ , then  $c \preceq_\mu B_j$  for some  $j \in 1..n$ . Moreover, by invertibility of subtyping of non-union types,  $B_j = c$ . On the other hand, by Lem. 3.38 (ii),  $A = c$ . Then,  $A \parallel_\epsilon \cap B \parallel_\epsilon \neq \emptyset$  and we conclude since  $\text{mmpos}(p, q) = \{\epsilon\}$ .
- $p = p_1 p_2$ : again, let's see the cases where  $p \not\triangleleft q$ . By matching success we have  $u = u_1 u_2$  a data structure with  $\llbracket u/p \rrbracket = \llbracket u_1/p_1 \rrbracket \uplus \llbracket u_2/p_2 \rrbracket$  both successful. Moreover, by Lem. 3.42,  $\exists D', B'$  such that  $D' @ B' \preceq_\mu B$  with  $\Gamma \triangleright u_1 : D'$  and  $\Gamma \triangleright u_2 : B'$ . Now we analyze the shape of  $q$ :
  - (i)  $q = y$ : as before, assume  $B = \bigoplus_{i \in 1..n} B_i$  with  $B_i \neq \oplus$  for every  $i \in 1..n$ . Then, by definition and invertibility of subtyping for non-union types, from  $D' @ B' \preceq_\mu B$  we have  $B_j = D'_j @ B'_j$  for some  $j \in 1..n$ . Again, by Lem. 3.38 (iii),  $\exists D, A'$  such that  $A = D @ A'$  and we conclude with  $A \parallel_\epsilon \cap B \parallel_\epsilon \neq \emptyset$ , given that  $\text{mmpos}(p, q) = \{\epsilon\}$ .
  - (ii)  $q = d$ : by Lem. 3.38 (ii) we have  $B = d$  which leads to a contradiction with  $D' @ B' \preceq_\mu B$ . Hence, this case is not possible.
  - (iii)  $q = q_1 q_2$ : by Lem. 3.38 (iii),  $\exists D'', B''$  such that  $B = D'' @ B''$  with  $\theta' \triangleright_p q_1 : D''$  and  $\theta' \triangleright_p q_2 : B''$ . Then, by invertibility of subtyping for non-union types, we get  $D' \preceq_\mu D''$  and  $B' \preceq_\mu B''$ . Thus,  $\Gamma \triangleright u_1 : D''$  and  $u_2 \triangleright B''$  by subsumption. On the other hand, by Lem. 3.38 (iii),  $\exists D, A'$  such that  $A = D @ A'$  with  $\theta \triangleright_p p_1 : D$  and  $\theta \triangleright_p p_2 : A'$ . Then, by inductive hypothesis, both  $\mathcal{P}_{\text{comp}}(p_1 : D', q_1 : D'')$  and  $\mathcal{P}_{\text{comp}}(p_2 : A', q_2 : B'')$  hold. Finally, since both patterns are compounds every mismatching position is internal, thus we can assure that  $\mathcal{P}_{\text{comp}}(p : A, q : B)$  holds too.

□

Let  $\Gamma, \theta$  be typing contexts,  $\sigma$  a substitution. We write  $\Gamma \vdash \sigma : \theta$  to indicate that  $\text{dom}(\sigma) = \text{dom}(\theta)$  and  $\Gamma \vdash \sigma(x) : \theta(x)$ , for all  $x \in \text{dom}(\sigma)$ . Likewise we use  $\Gamma \triangleright \sigma : \theta$  if each judgment is derivable. Two auxiliary results before addressing SR.

The following lemma assures that the substitution yielded by a successful match preserves the types of the variables in the pattern.

**Lemma 3.45 (Type of Successful Match)** *Suppose  $\llbracket u/p \rrbracket = \sigma$  is successful,  $\text{dom}(\theta) = \text{fm}(p)$ ,  $\theta \triangleright_p p : A$  and  $\Gamma \triangleright u : A$ . Then  $\Gamma \triangleright \sigma : \theta$ .*

**Proof.** By induction on  $p$ .

- $p = x$ . Then  $\sigma = \{u/x\}$  and, by Lem. 3.38 (i),  $x : A \in \theta$ . Then  $\theta = \{x : A\}$  and  $\Gamma \triangleright \sigma : \theta$  that holds by hypothesis.
- $p = c$ . The property holds trivially as  $\text{dom}(\sigma) = \emptyset = \text{dom}(\theta)$ .
- $p = p_1 p_2$ . Then, as the matching was successful,  $u = u_1 u_2$  is a data structure and  $\sigma = \{\{u_1/p_1\}\} \uplus \{\{u_2/p_2\}\} = \sigma_1 \uplus \sigma_2$ . By Lem. 3.38 (iii),  $\exists D, A'$  such that  $A = D @ A'$ ,  $\theta \triangleright_p p_1 : D$  and  $\theta \triangleright_p p_2 : A'$ . Then, by Lem. 3.39 (iii),  $\theta_1 \triangleright_p p_1 : D$  and  $\theta_2 \triangleright_p p_2 : A'$  with  $\theta_1 = \theta|_{\text{fm}(p_1)}$  and  $\theta_2 = \theta|_{\text{fm}(p_2)}$ .

On the other hand, by Lem. 3.42,  $\exists D', A''$  such that  $D' @ A'' \preceq_\mu A$ ,  $\Gamma \triangleright u_1 : D'$  and  $\Gamma \triangleright u_2 : A''$ . From  $D' @ A'' \preceq D @ A' \simeq_\mu A$  we get, by Prop. 3.32,  $D' \preceq_\mu D$  and  $A'' \preceq_\mu A'$ . Then we can derive  $\Gamma \triangleright u_1 : D$  and  $\Gamma \triangleright u_2 : A'$  by applying (T-SUBS).

Finally we can apply the inductive hypothesis on both side of the derivation and we get  $\Gamma \triangleright \sigma_1 : \theta_1$  and  $\Gamma \triangleright \sigma_2 : \theta_2$ . As  $\sigma_1$  and  $\sigma_2$  are disjoint then  $\theta_1$  and  $\theta_2$  are as well, and we can assure that  $\Gamma \triangleright \sigma : \theta$ .

□

Finally, we recall to the standard Substitution Lemma for type systems. It may also be interpreted in the context of an abstraction. Given  $p \rightarrow_\theta s$ , where  $\theta$  has the type assignments for the variables in  $p$ , every substitution that preserves  $\theta$  will also preserve the type of  $s$  once  $\theta$  is abstracted.

**Lemma 3.46 (Substitution Lemma)** *Suppose  $\Gamma, \theta \triangleright s : A$  and  $\Gamma \triangleright \sigma : \theta$ . Then  $\Gamma \triangleright \sigma s : A$ .*

**Proof.** By induction on  $s$ .

- $s = x$ . By Lem. 3.40 (i),  $\exists A'$  such that  $A' \preceq_\mu A$  and  $x : A' \in \Gamma, \theta$ . If  $x \in \text{dom}(\sigma)$ , as  $\text{dom}(\sigma) = \text{dom}(\theta)$ ,  $x : A' \in \theta$  and by hypothesis  $\Gamma \triangleright \sigma(x) : \theta(x)$ . Then by (T-SUBS) we get  $\Gamma \triangleright \sigma x : A$ . If not,  $x : A' \in \Gamma$  and  $\sigma x = x$ , then by (T-VAR) and (T-SUBS) we conclude  $\Gamma \triangleright \sigma x : A$ .
- $s = c$ . By Lem. 3.40 (ii),  $A \preceq_\mu c$  and, as  $\sigma c = c$ , by (T-CONST) and (T-SUBS) we have  $\Gamma \triangleright \sigma c : A$ .
- $s = r u$ . By Lem. 3.40 (iii) we have two cases:
  - (a) either  $\exists D, A'$  such that  $D @ A' \preceq_\mu A$ ,  $\Gamma, \theta \triangleright r : D$  and  $\Gamma, \theta \triangleright u : A'$ . By inductive hypothesis  $\Gamma \triangleright \sigma r : D$  and  $\Gamma \triangleright \sigma u : A'$ . As  $\sigma r \sigma u = \sigma(r u)$  by (T-COMP) and (T-SUBS) we get  $\Gamma \triangleright \sigma(r u) : A$ .
  - (b) or  $\exists A_1, \dots, A_n, A'$  such that  $A' \preceq_\mu A$ ,  $\Gamma, \theta \triangleright r : \bigoplus_{i \in 1..n} A_i \supset A'$ , and  $\Gamma, \theta \triangleright u : A_j$  for some  $j \in 1..n$ . Similarly to the previous case, we apply the inductive hypothesis to get  $\Gamma \triangleright \sigma r : \bigoplus_{i \in 1..n} A_i \supset A'$  and  $\Gamma \triangleright \sigma u : A_j$ . Then we conclude by (T-APP) and (T-SUBS) that  $\Gamma \triangleright \sigma(r u) : A$ .
- $s = (p_i \rightarrow_{\theta_i} s_i)_{i \in 1..n}$ . By Lem. 3.40 (iv),  $\exists A_1, \dots, A_n, B$  such that  $\bigoplus_{i \in 1..n} A_i \supset B \preceq_\mu A$ ,  $[p_i : A_i]_{i \in 1..n}$  is compatible,  $\text{dom}(\theta_i) = \text{fm}(p_i)$ ,  $\theta_i \triangleright_p p_i : A_i$  and  $\Gamma, \theta, \theta_i \triangleright s_i : B$  for every  $i \in 1..n$ . Without loss of generality we can assume  $\sigma$  avoids  $\theta_i$ <sup>9</sup> and  $\Gamma, \theta_i$  is a basis. Then  $\sigma s = (p_i \rightarrow_{\theta_i} \sigma s_i)_{i \in 1..n}$  and, by

<sup>9</sup> Here we mean  $\sigma$  avoids  $x$  for every  $x \in \text{dom}(\theta_i)$ .



Lem. 3.41 (i),  $\Gamma, \theta_i \triangleright \sigma : \theta$ . By inductive hypothesis we get  $\Gamma, \theta_i \triangleright \sigma s_i : B$  for every  $i \in 1..n$ . Finally, by (T-ABS) and (T-SUBS), we conclude  $\Gamma \triangleright \sigma s : A$ .  $\square$

## 4 Safety

Subject Reduction (Prop. 4.1) and Progress (Prop. 4.3) are addressed next.

**Proposition 4.1 (Subject Reduction)** *If  $\Gamma \triangleright s : A$  and  $s \rightarrow s'$ , then  $\Gamma \triangleright s' : A$ .*

**Proof.** By induction on  $s$ .

- $s = x$  or  $s = c$ . The property holds trivially as there is no  $s'$  such that  $s \rightarrow s'$ .
- $s = r u$ . Here we may consider three possibilities:
  - (i)  $r \rightarrow r'$ . By Lem. 3.40 (iii) we have two cases:
    - (a) either  $\exists D, A'$  such that  $D @ A' \preceq_\mu A$ ,  $\Gamma \triangleright r : D$  and  $\Gamma \triangleright u : A'$ . By inductive hypothesis  $\Gamma \triangleright r' : D$ . Then, by (T-COMP) and (T-SUBS), we have  $\Gamma \triangleright s' : A$ .
    - (b) or  $\exists A_1, \dots, A_n, A'$  such that  $A' \preceq_\mu A$ ,  $\Gamma \triangleright r : \bigoplus_{i \in 1..n} A_i \supset A'$ , and  $\Gamma \triangleright u : A_k$  for some  $k \in 1..n$ . By inductive hypothesis  $\Gamma \triangleright r' : \bigoplus_{i \in 1..n} A_i \supset A'$  and by applying (T-APP) and (T-SUBS) we conclude  $\Gamma \triangleright s' : A$ .
  - (ii)  $u \rightarrow u'$ . This case is similar to the previous one as by Lem. 3.40 we have the same two possible cases:
    - (a) either  $\exists D, A'$  such that  $D @ A' \preceq_\mu A$ ,  $\Gamma \triangleright r : D$  and  $\Gamma \triangleright u : A'$ . By inductive hypothesis  $\Gamma \triangleright u' : A'$ . Then, by (T-COMP) and (T-SUBS), we have  $\Gamma \triangleright s' : A$ .
    - (b) or  $\exists A_1, \dots, A_n, A'$  such that  $A' \preceq_\mu A$ ,  $\Gamma \triangleright r : \bigoplus_{i \in 1..n} A_i \supset A'$ , and  $\Gamma \triangleright u : A_k$  for some  $k \in 1..n$ . By inductive hypothesis  $\Gamma \triangleright u' : A_k$  and by applying (T-APP) and (T-SUBS) we conclude  $\Gamma \triangleright s' : A$ .
  - (iii)  $r = (p_i \rightarrow_{\theta_i} s_i)_{i \in 1..n}$  and  $s' = \{\{u/p_k\}\} s_k$  for some  $k \in 1..n$  such that  $\{\{u/p_k\}\} = \sigma$  and  $\{\{u/p_i\}\} = \mathbf{fail}$  for every  $i < k$ . Assume, towards an absurd, that Lem. 3.40 (iii.a) holds for  $s$ . Then,  $\exists D, A'$  such that  $D @ A' \preceq_\mu A$ ,  $\Gamma \triangleright r : D$  and  $\Gamma \triangleright u : A'$ . But, by Lem. 3.40 (iv) applied to  $\Gamma \triangleright r : D$ ,  $\exists A_1, \dots, A_n, B$  such that  $\bigoplus_{i \in 1..n} A_i \supset B \preceq_\mu D$  and, by Lem. 3.34,  $\exists \mathcal{U}$  such that  $D \simeq_\mu \mathcal{U}[B']$  with  $\bigoplus_{i \in 1..n} A_i \supset B \preceq_\mu B'$  which is a contradiction since  $D$  is a data type. Thus, it must be the case that Lem. 3.40 (iii.b) holds for  $s$ .

Then,  $\exists C_1, \dots, C_m, A'$  such that  $A' \preceq_\mu A$ ,  $\Gamma \triangleright r : \bigoplus_{j \in 1..m} C_j \supset A'$  and:

$$\Gamma \triangleright u : C_{k'} \tag{11}$$

for some  $k' \in 1..m$ . Applying once again Lem. 3.40 (iv), this time to  $\Gamma \triangleright r : \bigoplus_{j \in 1..m} C_j \supset A'$ , we get  $\exists A_1, \dots, A_n, B$  such that:

$$\bigoplus_{i \in 1..n} A_i \supset B \preceq_\mu \bigoplus_{j \in 1..m} C_j \supset A' \tag{12}$$

$\text{dom}(\theta_i) = \text{fm}(p_i)$ ,  $[p_i : A_i]_{i \in 1..n}$  is compatible,  $\theta_i \triangleright_p p_i : A_i$  and  $\Gamma, \theta_i \triangleright s_i : B$  for every  $i \in 1..n$ .



From (12) and Prop. 3.32 we have  $B \preceq_\mu A'$  and

$$\bigoplus_{j \in 1..m} C_j \preceq_\mu \bigoplus_{i \in 1..n} A_i \quad (13)$$

We want to show that  $\Gamma \triangleright u : A_k$ . For that we need to distinguish two cases:

(a) If  $u$  is in matchable form, we have two possibilities:

$u$  is a data structure: then, by Lem. 3.42, there exists a non-union datatype  $D$  such that  $D \preceq_\mu C_{k'}$  and  $\Gamma \triangleright u : D$ .

$u$  is an abstraction: then, by Lem. 3.40 (iv),  $\exists C', C''$  such that  $C' \supset C'' \preceq_\mu C_{k'}$  and  $\Gamma \triangleright u : C' \supset C''$ .

Then, in both cases there exists a non-union type, say  $C$ , such that  $C \preceq_\mu C_{k'}$  and  $\Gamma \triangleright u : C$ . Then, from (13) we get:

$$C \preceq_\mu \bigoplus_{i \in 1..n} A_i$$

and, since  $C$  is non-union,  $C \preceq_\mu A_l$  for some  $l \in 1..n$ . Hence, by subsumption  $\Gamma \triangleright u : A_l$ .

If  $k = l$  we are done, so assume  $k \neq l$ . Recall the conditions for the reduction rule, where  $\llbracket u/p_i \rrbracket = \mathbf{fail}$  for every  $i < k$ . Then, by Lem. 3.43, we have  $A_l \not\preceq_\mu A_i$ . Thus, it must be the case that  $k < l$ . By Lem. 3.44 with hypothesis  $\Gamma \triangleright u : A_l$ ,  $\theta_k \triangleright_p p_k : A_k$ ,  $\theta_l \triangleright_p p_l : A_l$  and  $\llbracket u/p_k \rrbracket = \sigma$  we get that  $\mathcal{P}_{\text{comp}}(p_k : A_k, p_l : A_l)$  holds. Additionally, we already saw that the list  $[p_i : A_i]_{i \in 1..n}$  is compatible, thus  $p_k : A_k \lll p_l : A_l$  and by definition  $A_l \preceq_\mu A_k$ . Finally we conclude by subsumption once again,  $\Gamma \triangleright u : A_k$ .

(b) If  $u$  is not in matchable form, then  $p_k = x$  and by the premises of the reductions rule we need  $\llbracket u/p_i \rrbracket = \mathbf{fail}$  for every  $i < k$ . Thus, necessarily  $k = 1$ . Moreover, since  $x \triangleleft p_i$  for every  $i \in 1..n$ , by compatibility we have  $A_i \preceq_\mu A_k$ . Then, from (13) we get

$$C_{k'} \preceq_\mu \bigoplus_{j \in 1..m} C_j \preceq_\mu \bigoplus_{i \in 1..n} A_i \preceq_\mu A_k$$

Thus, by subsumption,  $\Gamma \triangleright u : A_k$ .

Finally, in either case we have  $\Gamma \triangleright u : A_k$ . Now Lem. 3.45 and 3.46 with  $\Gamma, \theta_k \triangleright s_k : B$  entails  $\Gamma \triangleright s' : B$  and we conclude by subsumption,  $\Gamma \triangleright s' : A$  (recall  $B \preceq_\mu A' \preceq_\mu A$ ).

- $s = (p_i \rightarrow_{\theta_i} s_i)_{i \in 1..n}$ . Then  $s' = p_1 \rightarrow_{\theta_1} s_1 \mid \dots \mid p_k \rightarrow_{\theta_k} s'_k \mid \dots \mid p_n \rightarrow_{\theta_n} s_n$  with  $s_k \rightarrow s'_k$ . By Lem. 3.40 (iv),  $\exists A_1, \dots, A_n, B$  s.t.  $\bigoplus_{i \in 1..n} A_i \supset B \preceq_\mu A$ ,  $[p_i : A_i]_{i \in 1..n}$  is compatible,  $\text{dom}(\theta_i) = \text{fm}(p_i)$ ,  $\theta_i \triangleright_p p_i : A_i$  and  $\Gamma, \theta_i \triangleright s_i : B$  for every  $i \in 1..n$ . By inductive hypothesis  $\Gamma, \theta_k \triangleright s'_k : A_k$  and by applying (T-ABS) and (T-SUBS) we conclude  $\Gamma \triangleright s' : A$ .

□

Let the set of **values** be defined as  $v ::= x v_1 \dots v_n \mid \mathbf{c} v_1 \dots v_n \mid (p_i \rightarrow_{\theta_i} s_i)_{i \in 1..n}$ . The following auxiliary property guarantees the success of matching for well-typed closed values (note that values are already in matchable form).

**Lemma 4.2 (Successful Match for Closed Values)** *Suppose  $\triangleright v : A$  and  $\theta \triangleright_p p : A$  where  $v$  is a value. Then,  $\llbracket v/p \rrbracket$  is successful.*

**Proof.** By induction on  $p$ . Note that  $v$  cannot be a variable since it is typed on the empty context and, by Lem. 3.41,  $\text{fv}(v) \subseteq \emptyset$ . Hence, it is a closed term. Then  $v$  is either a data structure or a case.

- $p = x$ . The property holds trivially with the substitution  $\{v/x\}$ .
- $p = \mathbf{c}$ . By Lem. 3.38 (ii),  $A = \mathbf{c}$ . Suppose  $v = (q_i \rightarrow_{\theta_i} s_i)_{i \in 1..n}$ . By Lem. 3.40 (iv),  $\exists A_1, \dots, A_n, B$  such that  $\bigoplus_{i \in 1..n} A_i \supset B \preceq_{\mu} \mathbf{c}$  and, by Lem. 3.34,  $\exists \mathcal{U}, A'$  such that  $\mathbf{c} \simeq_{\mu} \mathcal{U}[A']$ ,  $\bigoplus_{i \in 1..n} A_i \supset B \preceq_{\mu} A'$  and they both have the same outermost type constructor. This leads to a contradiction. Hence  $v$  is not a case.

Then it must be a data structure. By Lem. 3.42,  $\exists D$  such that  $D$  is a non-union type,  $D \preceq_{\mu} \mathbf{c}$  and  $\triangleright r : D$ . Furthermore, case (2) of the lemma does not hold since  $A \simeq_{\mu} \mathbf{c}$ . Then, by case (1),  $v = \mathbf{c}$  and  $D \simeq_{\mu} \mathbf{c}$ . Finally we can assure that  $\llbracket v/p \rrbracket = \llbracket \mathbf{c}/\mathbf{c} \rrbracket$  is successful.

- $p = p_1 p_2$ . By Lem. 3.38 (iii),  $\exists D, A'$  such that  $A = D @ A'$ ,  $\theta \triangleright_p p_1 : D$  and  $\theta \triangleright_p p_2 : A'$ . Similarly to the previous case we may conclude that if  $v = (q_i \rightarrow_{\theta_i} s_i)_{i \in 1..n}$  there exists a functional type  $B$  such that  $D @ A' \simeq_{\mu} \mathcal{U}[B]$  which leads to a contradiction. Hence we are again in the case that  $v$  is a data structure.

By Lem. 3.42,  $\exists D'$  such that  $D'$  is a non-union type,  $D' \preceq_{\mu} D @ A'$  and  $\triangleright v : D'$ . Moreover, we can assure that case (2) of the lemma holds, so we have  $v = v_1 v_2$  and  $\exists D'', A''$  such that  $D' \simeq_{\mu} D'' @ A''$ ,  $\triangleright v_1 : D''$  and  $\triangleright v_2 : A''$ . Now by Prop. 3.32 with  $D'' @ A'' \preceq_{\mu} D @ A'$  we get  $D'' \preceq_{\mu} D$  and  $A'' \preceq_{\mu} A'$ , and by (T-SUBS)  $\triangleright v_1 : D$  and  $\triangleright v_2 : A'$ .

Then we can apply the inductive hypothesis and to deduce that both  $\llbracket v_1/p_1 \rrbracket$  and  $\llbracket v_2/p_2 \rrbracket$  are successful. Finally by linearity of patterns we can safely conclude that  $\llbracket v/p \rrbracket = \llbracket v_1/p_1 \rrbracket \uplus \llbracket v_2/p_2 \rrbracket$  is also successful.  $\square$

**Proposition 4.3 (Progress)** *If  $\triangleright s : A$  and  $s$  is not a value, then  $\exists s'$  s.t.  $s \rightarrow s'$ .*

**Proof.** By induction on  $s$  analyzing the subterm of  $s$  that is not yet a value.

- $s = x$ ,  $s = \mathbf{c}$  or  $s = (p_i \rightarrow_{\theta_i} s_i)_{i \in 1..n}$ . The property holds trivially as  $s$  is already a value.
- $s = r u$ . Here we have three possible cases:
  - (i)  $r$  is not yet a value. Then, by Lem. 3.40 (iii),  $\exists A_1, A_2$  such that  $\triangleright r : A_1$  and  $\triangleright u : A_2$ . By inductive hypothesis  $\exists r'$  such that  $r \rightarrow r'$  and we conclude with  $s' = r' u$ .
  - (ii)  $r$  is a value and  $u$  is not. Again by Lem. 3.40 (iii),  $\exists A_1, A_2$  such that  $\triangleright r : A_1$  and  $\triangleright u : A_2$ . By inductive hypothesis  $\exists u'$  such that  $u \rightarrow u'$  and we conclude with  $s' = r u'$ .
  - (iii)  $r = (p_i \rightarrow_{\theta_i} s_i)_{i \in 1..n}$  with  $u$  already a value. As for SR, by Lem. 3.40 (iii.b), we have that  $\exists C_1, \dots, C_m, A'$  such that  $A' \preceq_{\mu} A$ ,  $\triangleright r : \bigoplus_{j \in 1..m} C_j \supset A'$  and

$$\triangleright u : C_{k'} \tag{14}$$

for some  $k' \in 1..m$ . And, by Lem. 3.40 (iv) on  $\triangleright r : \bigoplus_{j \in 1..m} C_j \supset A'$ ,

$\exists A_1, \dots, A_n, B$  such that

$$\bigoplus_{i \in 1..n} A_i \supset B \preceq_\mu \bigoplus_{j \in 1..m} C_j \supset A' \quad (15)$$

$\text{dom}(\theta_i) = \text{fm}(p_i), [p_i : A_i]_{i \in 1..n}$  is compatible,  $\theta_i \triangleright_p p_i : A_i$  and  $\theta_i \triangleright s_i : B$  for every  $i \in 1..n$ .

From (15) and Prop. 3.32 we have  $B \preceq_\mu A'$  and

$$\bigoplus_{j \in 1..m} C_j \preceq_\mu \bigoplus_{i \in 1..n} A_i \quad (16)$$

Additionally, by (14) and Lem. 3.41 we know that  $u$  is a closed value, *i.e.* a data structure or an abstraction. Hence,  $u$  is in matchable form and matching against every pattern  $p_i$  is decided. Then, we have to possibilities as in the proof for SR:

- (a)  $u$  is a data structure: by Lem. 3.42, there exists a non-union datatype  $D$  such that  $D \preceq_\mu C_{k'}$  and  $\Gamma \triangleright u : D$ .
- (b)  $u$  is an abstraction: by Lem. 3.40 (iv),  $\exists C', C''$  such that  $C' \supset C'' \preceq_\mu C_{k'}$  and  $\Gamma \triangleright u : C' \supset C''$ .

In both cases we can assume there is a non-union type, say  $C$ , such that  $C \preceq_\mu C_{k'}$  and  $\triangleright u : C$ . Then, from (16) we get  $C \preceq_\mu \bigoplus_{i \in 1..n} A_i$  and  $C \preceq_\mu A_k$  for some  $k \in 1..n$ , as before. Thus, by subsumption,  $\triangleright u : A_k$ . Finally, with  $\theta_k \triangleright_p p_k : A_k$  we are under the hypothesis of Lem. 4.2, and we conclude by taking  $s' = \llbracket u/p_k \rrbracket s_k$ .

□

## 5 Conclusions

A type system is proposed for a calculus that supports path polymorphism and two fundamental properties are addressed, namely Subject Reduction and Progress. The type system includes type application, constants as types, union and recursive types. Both properties rely crucially on a notion of pattern *compatibility* and on invertibility of subtyping of  $\mu$ -types. This last result is proved via a coinductive semantics for the finite  $\mu$ -types. Regarding future work an outline of possible avenues follows.

- There exists extensive work on type-checking for recursive types [1, 20, 26], including some efficient algorithms for both equivalence [23] and subtyping [21]. We are currently adapting these ideas to CAP.
- We already mentioned the addition of parametric polymorphism (presumably in the style of  $F_{<}$ : [8, 15, 25]). We believe this should not present major difficulties.
- Strong normalization requires devising a notion of positive/negative occurrence in the presence of strong  $\mu$ -type equality, which is known not to be obvious [4, page 515].
- A more ambitious extension is that of *dynamic patterns*, namely patterns that may be computed at run-time, PPC being the prime example of a calculus supporting this feature.

## References

- [1] Amadio, R. M. and L. Cardelli, *Subtyping recursive types*, ACM Trans. Program. Lang. Syst. **15** (1993), pp. 575–631.
- [2] Arbiser, A., A. Miquel and A. Ríos, *The lambda-calculus with constructors: Syntax, confluence and separation*, J. Funct. Program. **19** (2009), pp. 581–631.
- [3] Ariola, Z. M. and J. W. Klop, *Equational term graph rewriting*, Fundam. Inform. **26** (1996), pp. 207–240.
- [4] Barendregt, H., W. Dekkers and R. Statman, editors, “Lambda Calculus with Types,” Perspectives in Logic, CUP, 2013.
- [5] Barthe, G., H. Cirstea, C. Kirchner and L. Liquori, *Pure patterns type systems*, in: A. Aiken and G. Morrisett, editors, *Conference Record of POPL 2003: The 30th SIGPLAN-SIGACT Symposium on Principles of Programming Languages, New Orleans, Louisiana, USA, January 15-17, 2003* (2003), pp. 250–261.  
URL <http://doi.acm.org/10.1145/640128.604152>
- [6] Bezem, M., J. W. Klop and R. de Vrijer, editors, “Term Rewriting Seminar – Terese,” Cambridge University Press, 2003.
- [7] Brandt, M. and F. Henglein, *Coinductive axiomatization of recursive type equality and subtyping*, Fundam. Inform. **33** (1998), pp. 309–338.
- [8] Cardelli, L., S. Martini, J. C. Mitchell and A. Scedrov, *An extension of System F with subtyping*, in: T. Ito and A. R. Meyer, editors, *Theoretical Aspects of Computer Software*, LNCS **526**, Springer Berlin Heidelberg, 1991 pp. 750–770.  
URL [http://dx.doi.org/10.1007/3-540-54415-1\\_73](http://dx.doi.org/10.1007/3-540-54415-1_73)
- [9] Cardone, F., *An algebraic approach to the interpretation of recursive types*, in: J.-C. Raoult, editor, *CAAP*, LNCS **581** (1992), pp. 66–85.
- [10] Cirstea, H. and C. Kirchner, *The rewriting calculus - part i and ii*, Logic Journal of the IGPL **9** (2001), pp. 339–410.
- [11] Cirstea, H., C. Kirchner and L. Liquori, *Matching power*, in: A. Middeldorp, editor, *Rewriting Techniques and Applications, 12th International Conference, RTA 2001, Utrecht, The Netherlands, May 22-24, 2001, Proceedings*, LNCS **2051** (2001), pp. 77–92.  
URL [http://dx.doi.org/10.1007/3-540-45127-7\\_8](http://dx.doi.org/10.1007/3-540-45127-7_8)
- [12] Cirstea, H., C. Kirchner and L. Liquori, *The rho cube*, in: F. Honsell and M. Miculan, editors, *Foundations of Software Science and Computation Structures, 4th International Conference, FOSSACS 2001 (ETAPS 2001), Genova, Italy, April 2-6, 2001, Proceedings*, LNCS **2030** (2001), pp. 168–183.  
URL [http://dx.doi.org/10.1007/3-540-45315-6\\_11](http://dx.doi.org/10.1007/3-540-45315-6_11)
- [13] Cirstea, H., C. Kirchner and L. Liquori, *Rewriting calculus with(out) types*, Electr. Notes Theor. Comput. Sci. **71** (2002), pp. 3–19.  
URL [http://dx.doi.org/10.1016/S1571-0661\(05\)82526-5](http://dx.doi.org/10.1016/S1571-0661(05)82526-5)
- [14] Cirstea, H., L. Liquori and B. Wack, *Rewriting calculus with fixpoints: Untyped and first-order systems*, in: S. Berardi, M. Coppo and F. Damiani, editors, *Types for Proofs and Programs, International Workshop, TYPES 2003, Torino, Italy, April 30 - May 4, 2003, Revised Selected Papers*, LNCS **3085** (2003), pp. 147–161.  
URL [http://dx.doi.org/10.1007/978-3-540-24849-1\\_10](http://dx.doi.org/10.1007/978-3-540-24849-1_10)
- [15] Colazzo, D. and G. Ghelli, *Subtyping recursion and parametric polymorphism in kernel fun*, Information and Computation **198** (2005), pp. 71–147.  
URL <http://www.sciencedirect.com/science/article/pii/S0890540105000167>
- [16] Courcelle, B., *Fundamental properties of infinite trees.*, Theor. Comput. Sci. **25** (1983), pp. 95–169.  
URL <http://dblp.uni-trier.de/db/journals/tcs/tcs25.html#Courcelle83>
- [17] Jay, C. B., “Pattern Calculus: Computing with Functions and Structures,” Springer, 2009.
- [18] Jay, C. B. and D. Kesner, *Pure pattern calculus*, in: P. Sestoft, editor, *ESOP*, LNCS **3924** (2006), pp. 100–114.
- [19] Jay, C. B. and D. Kesner, *First-class patterns*, J. Funct. Program. **19** (2009), pp. 191–225.
- [20] Jha, S., J. Palsberg and T. Zhao, *Efficient type matching*, in: M. Nielsen and U. Engberg, editors, *Foundations of Software Science and Computation Structures, 5th International Conference, FOSSACS 2002 (ETAPS 2002) Grenoble, France, April 8-12, 2002, Proceedings*, LNCS **2303** (2002), pp. 187–204.  
URL [http://dx.doi.org/10.1007/3-540-45931-6\\_14](http://dx.doi.org/10.1007/3-540-45931-6_14)

- [21] Kozen, D., J. Palsberg and M. I. Schwartzbach, *Efficient recursive subtyping*, Mathematical Structures in Computer Science **5** (1995), pp. 113–125.  
URL <http://dx.doi.org/10.1017/S0960129500000657>
- [22] Liquori, L. and B. Wack, *The polymorphic rewriting-calculus: [type checking vs. type inference]*, Electr. Notes Theor. Comput. Sci. **117** (2005), pp. 89–111.  
URL <http://dx.doi.org/10.1016/j.entcs.2004.06.027>
- [23] Palsberg, J. and T. Zhao, *Efficient and flexible matching of recursive types*, Inf. Comput. **171** (2001), pp. 364–387.  
URL <http://dx.doi.org/10.1006/inco.2001.3090>
- [24] Petit, B., *Semantics of typed lambda-calculus with constructors*, Logical Methods in Computer Science **7** (2011).
- [25] Pierce, B. C., “Types and Programming Languages,” MIT Press, Cambridge, MA, USA, 2002.
- [26] ten Eikelder, H., *Some algorithms to decide the equivalence of recursive types*, Technical Report 91/31, Eindhoven University of Technology (1991).
- [27] Vouillon, J., *Subtyping union types*, in: J. Marcinkowski and A. Tarlecki, editors, *CSL*, LNCS **3210** (2004), pp. 415–429.